Giới thiệu về mạng tích chập Conv Neural Networks

Mục lục

- Giới thiệu tổng quan
- Lịch sử CNN
- Các lớp trong mạng CNN
- Một vài mạng CNN cơ bản

Giới thiệu tổng quan

Tương quan và tích chập

- Tương quan 2D

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

`h=filter2(f,I);` or `h=imfilter(I,f);`

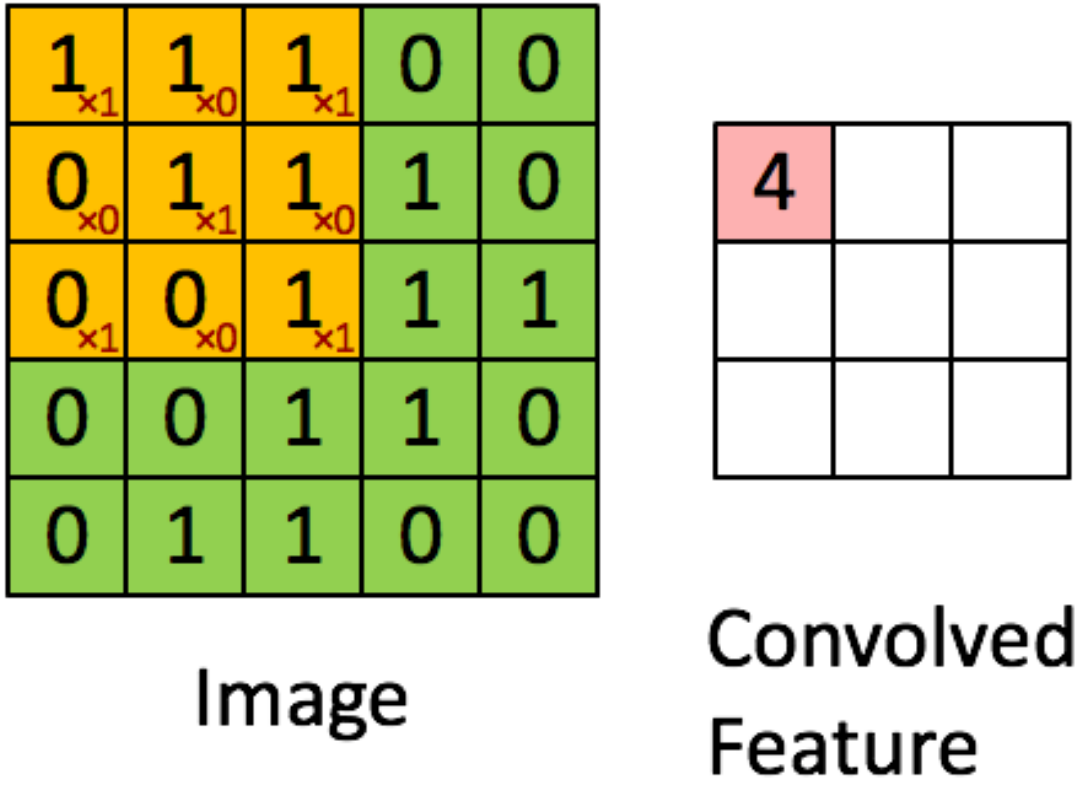
- Tích chập 2D

$$h[m,n] = \sum_{k,l} f[k,l]\, I[m-k, n-l]$$

`h=conv2(f,I);` or `h=imfilter(I,f,'conv');`

Tương quan và tích chập 2D giống nhau khi bộ lọc đối xứng

Cách thức hoạt động của bộ lọc



© http://deeplearning.stanford.edu/

Tích chập/tương quan trong xử lý ảnh

input

Kernel for blurring

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

tf.nn.conv2d

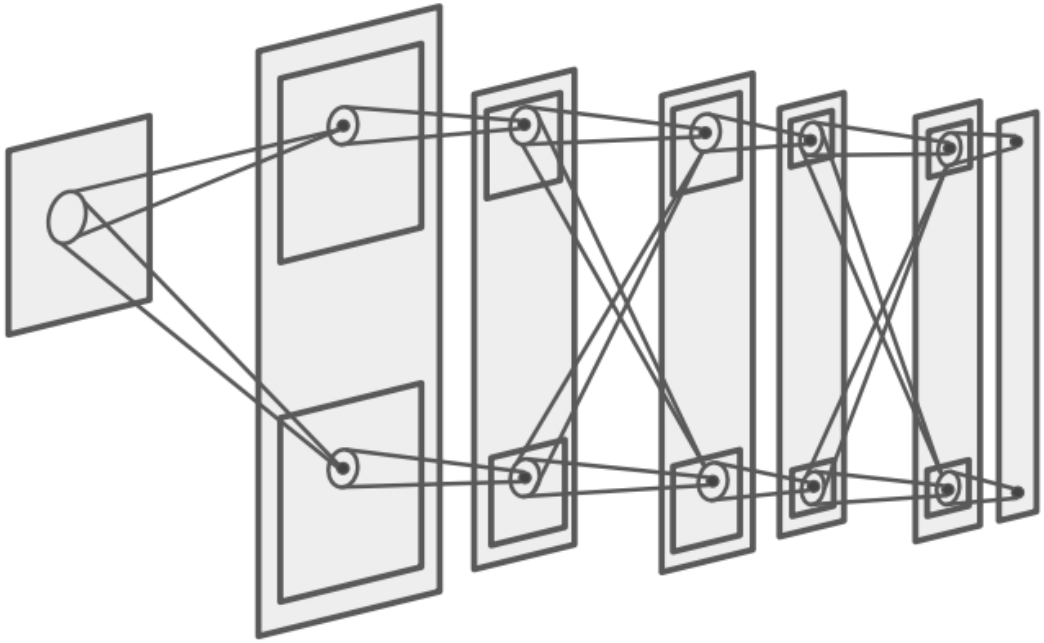
output

© http://web.stanford.edu/class/cs20si

Lịch sử CNN

Lịch sử CNNs

Neocognitron
[Fukushima 1980]



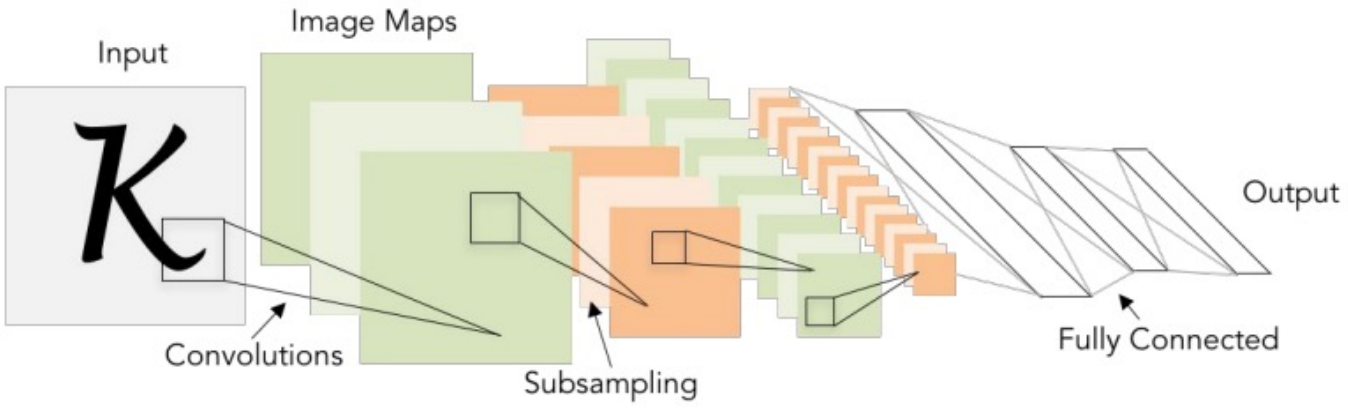
"sandwich" architecture (SCSCSC…)
simple cells: modifiable parameters
complex cells: perform pooling

- Ý tưởng CNNs xuất phát đầu tiên từ công trình của Fukushima năm 1980

Lịch sử CNNs

Gradient-based learning applied to document recognition

[LeCun, Bottou, Bengio, Haffner 1998]



- Năm 1998, LeCun áp dụng BackProp huấn luyện mạng CNNs cho bài toán nhận dạng văn bản

Lịch sử CNNs

ImageNet Classification with Deep Convolutional Neural Networks
[Krizhevsky, Sutskever, Hinton, 2012]

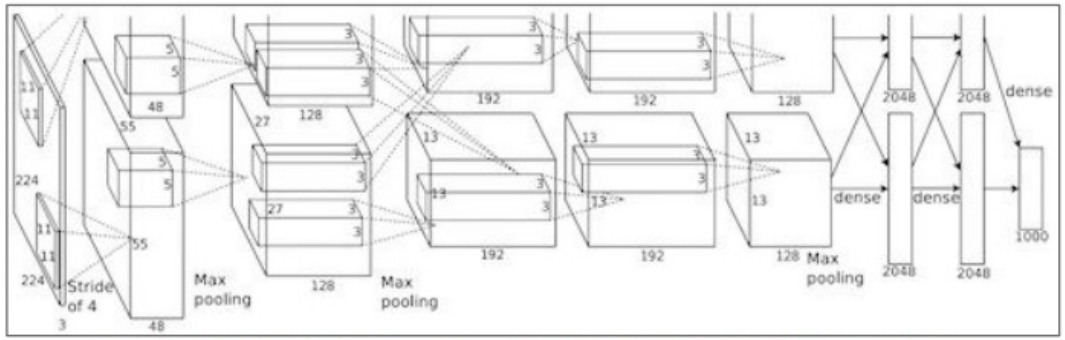
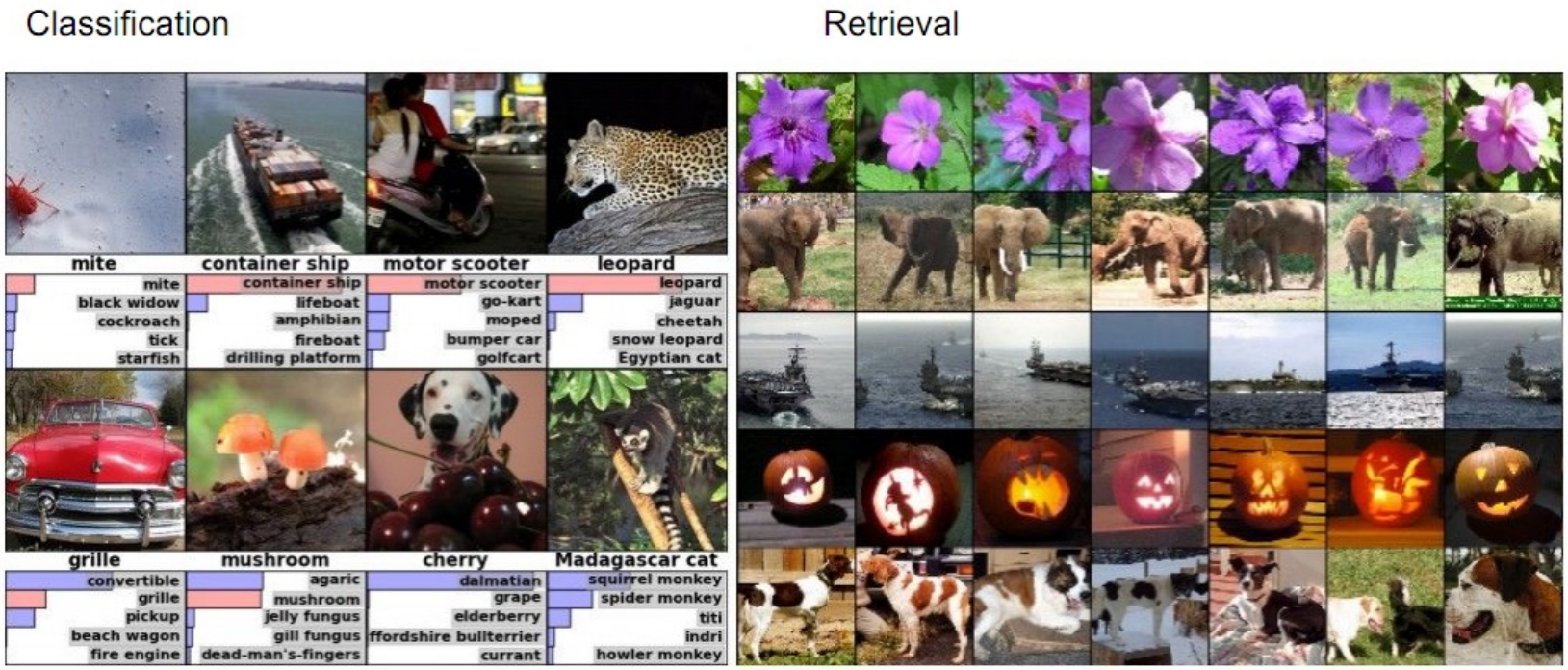


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

"AlexNet"

- Năm 2012, CNNs gây tiếng vang lớn khi vô địch cuộc thi ILSRC 2012, vượt xa phương pháp đứng thứ 2 theo cách tiếp cận thị giác máy tính truyền thống.

Lịch sử CNNs



Figures copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

- Hiện nay CNNs ứng dụng khắp nơi, ví dụ trong bài toán phân loại ảnh, truy vấn ảnh

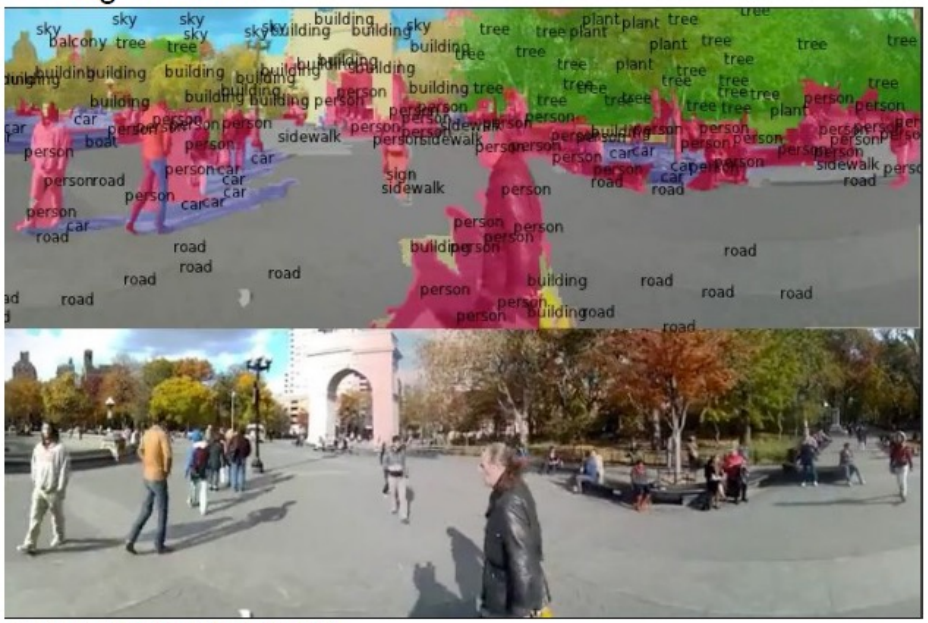
Lịch sử CNNs

Detection

Figures copyright Shaoqing Ren, Kaiming He, Ross Girschick, Jian Sun, 2015. Reproduced with permission.

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



Figures copyright Clement Farabet, 2012. Reproduced with permission.

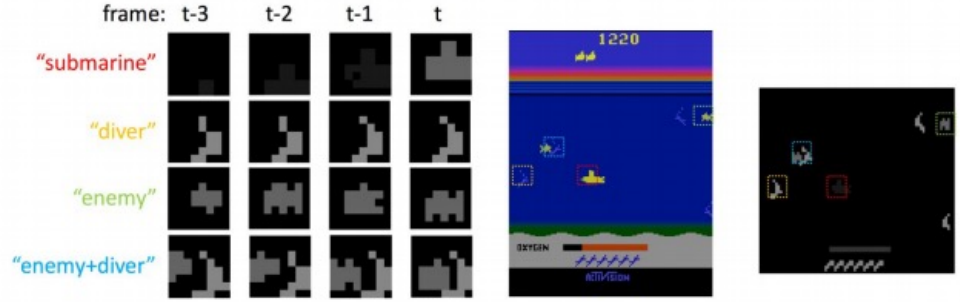
[Farabet et al., 2012]

- Ứng dụng CNNs trong bài toán phát hiện đối tượng, phân đoạn ảnh

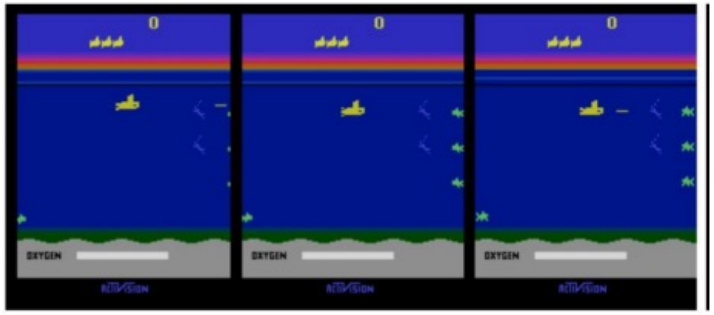
Lịch sử CNNs

Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014]



[Guo et al. 2014]



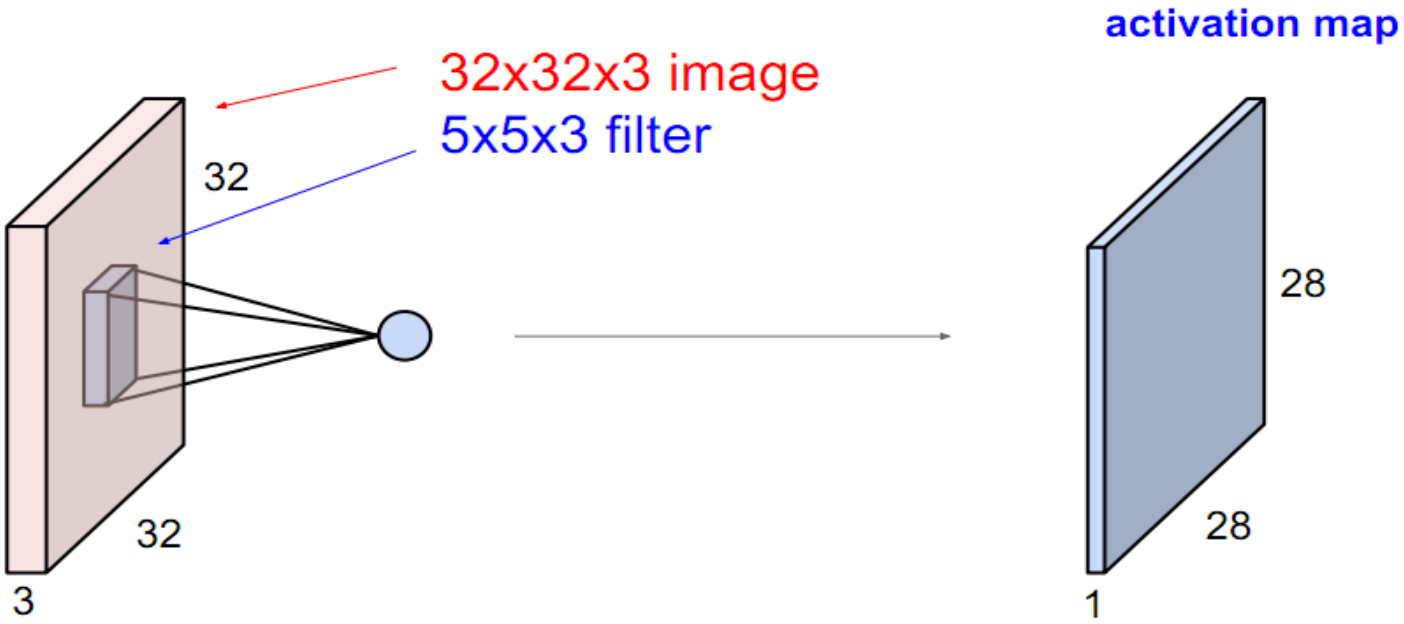
Figures copyright Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard Lewis, and Xiaoshi Wang, 2014. Reproduced with permission.

- Ứng dụng CNNs trong nhận dạng dáng người (human pose), trong trò chơi…

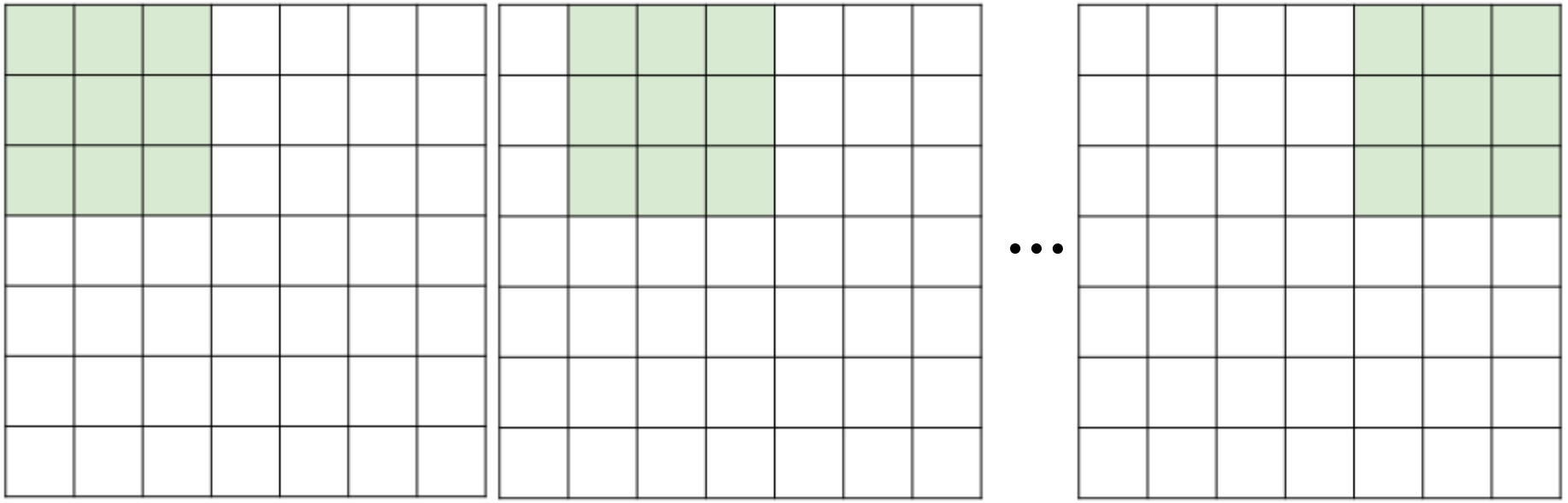
Các lớp trong mạng CNN

Lớp tích chập

- Khác với nơ-ron kết nối đầy đủ, mỗi nơ-ron tích chập (filter) chỉ kết nối cục bộ với dữ liệu đầu vào
- Nơ-ron tích chập trượt từ trái sang phải và từ trên xuống dưới khối dữ liệu đầu vào và tính toán để sinh ra một bản đồ kích hoạt (activation map)
- Chiều sâu của nơ-ron tích chập bằng chiều sâu của khối dữ liệu đầu vào

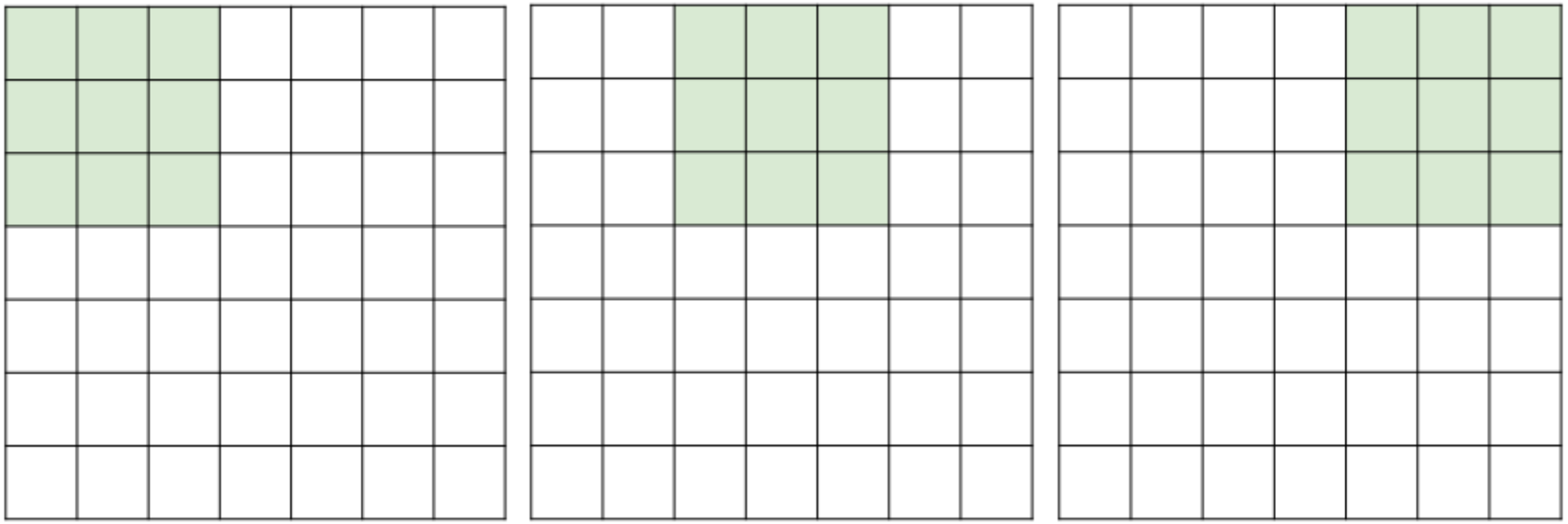


Lớp tích chập



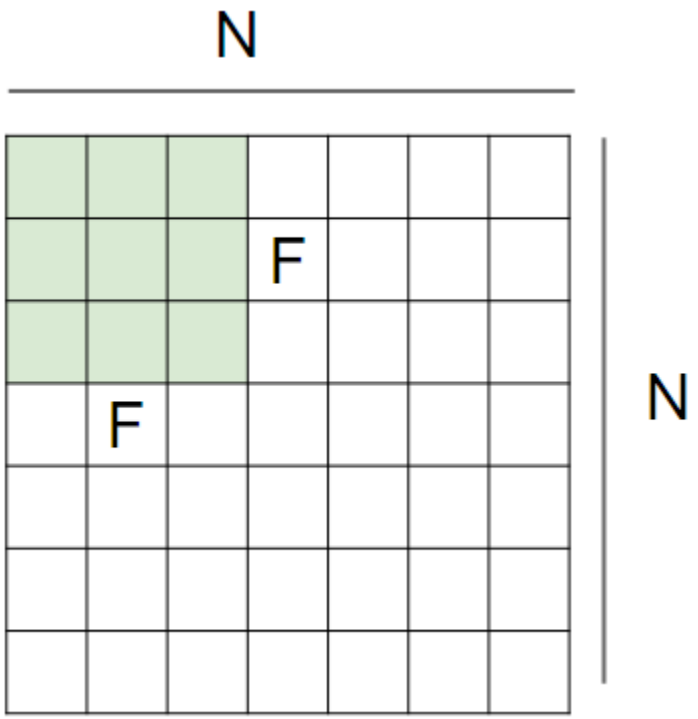
- Bước nhảy stride = 1
- Đầu vào kích thước 7x7, nơ-ron kích thước 3x3
- Đầu ra kích thước 5x5

Lớp tích chập



- Bước nhảy stride = 2
- Đầu vào kích thước 7x7, nơ-ron kích thước 3x3
- Đầu ra kích thước 3x3

Lớp tích chập

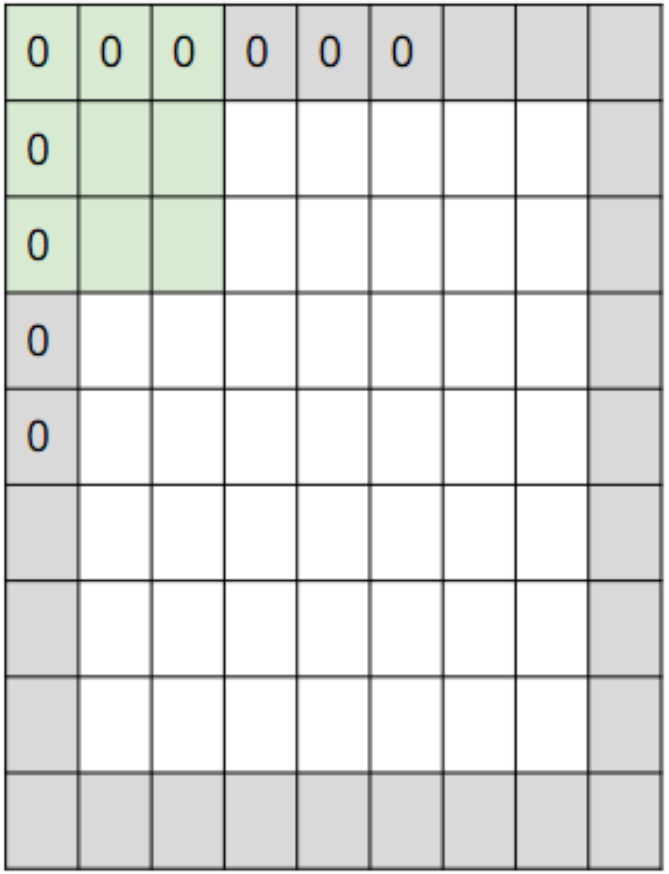


Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33 :\

Lớp tích chập

- Để bảo toàn kích thước thường thêm viền bởi các số 0 (zero padding).
- Ví dụ: đầu vào kích thước 7x7, nơ-ron kích thước 3x3, bước nhảy stride 1, padding viền độ rộng 1.
- Khi đó kích thước đầu ra là 7x7

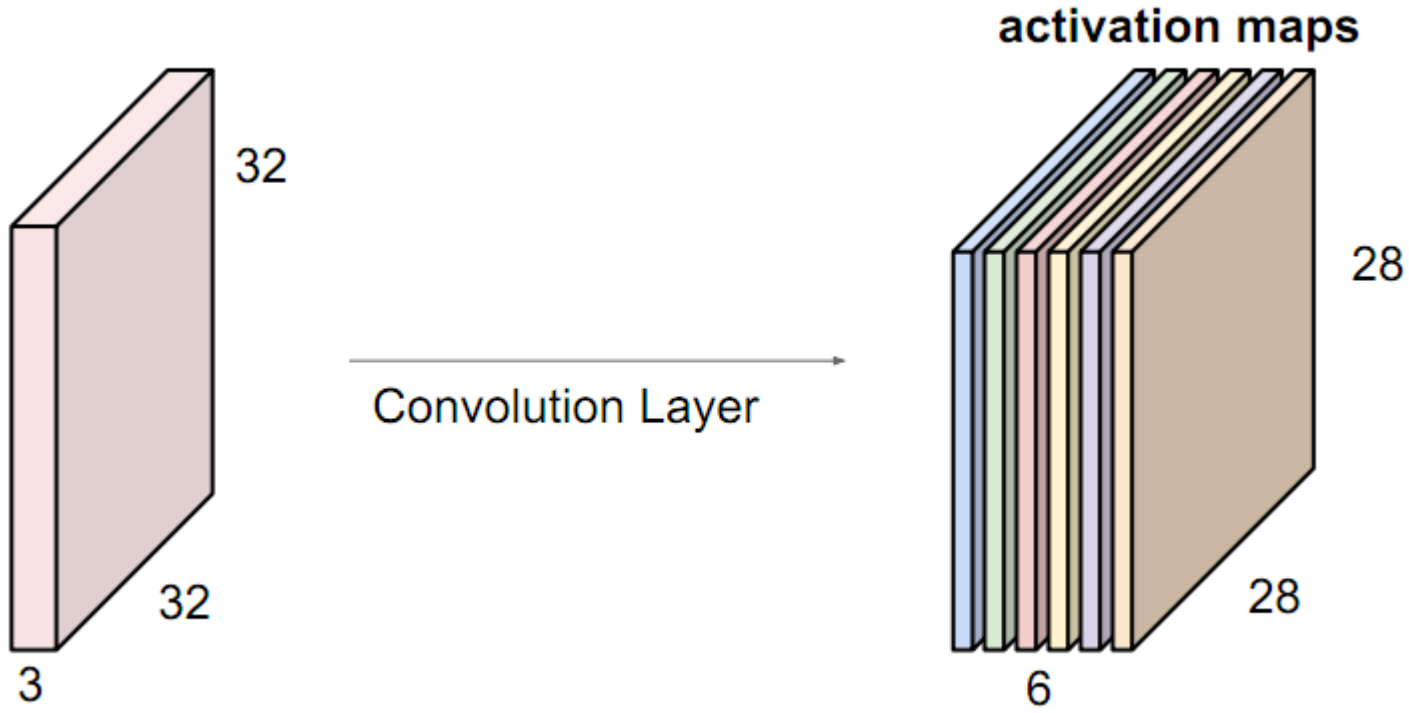


Lớp tích chập

- Giả sử có thêm nơ-ron tích chập khác thì nó cũng hoạt động tương tự và sinh ra bản đồ kích hoạt thứ hai
- Lưu ý trọng số của các nơ-ron tích chập là khác nhau

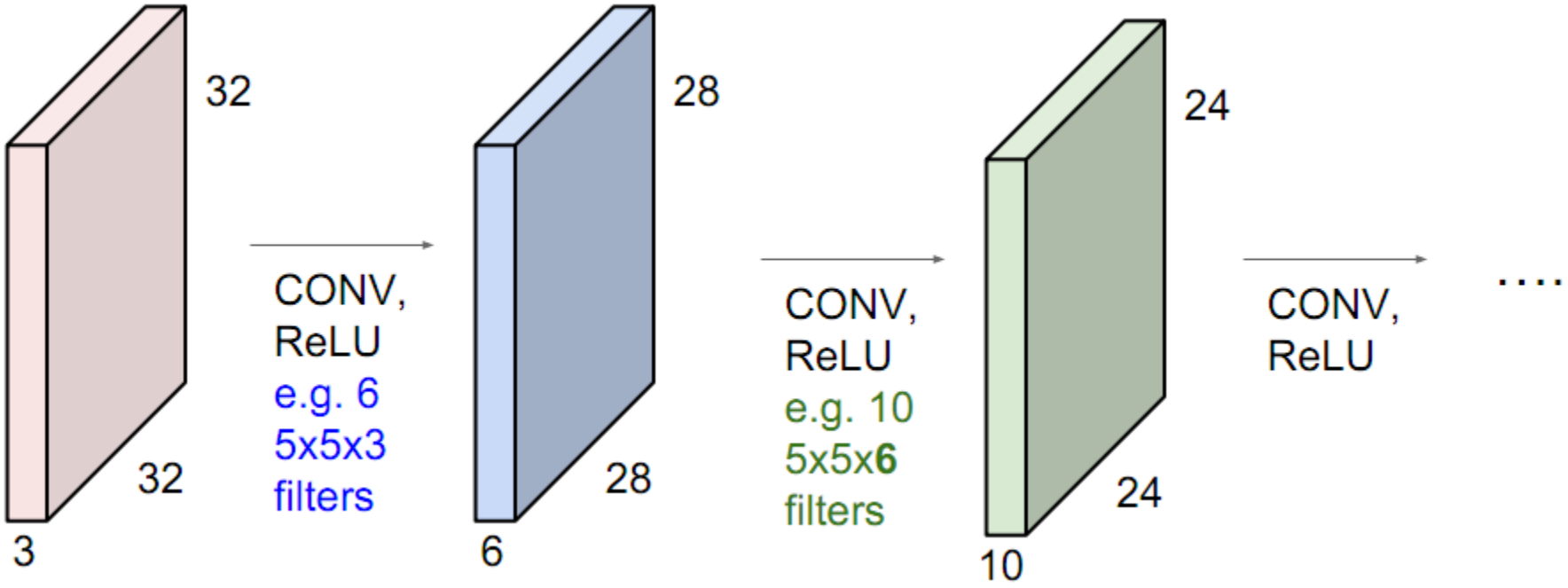
Lớp tích chập

- Giả sử có 6 nơ-ron tích chập sẽ sinh ra 6 bản đồ kích hoạt
- Các bản đồ kích hoạt ghép với nhau thành một "ảnh mới"



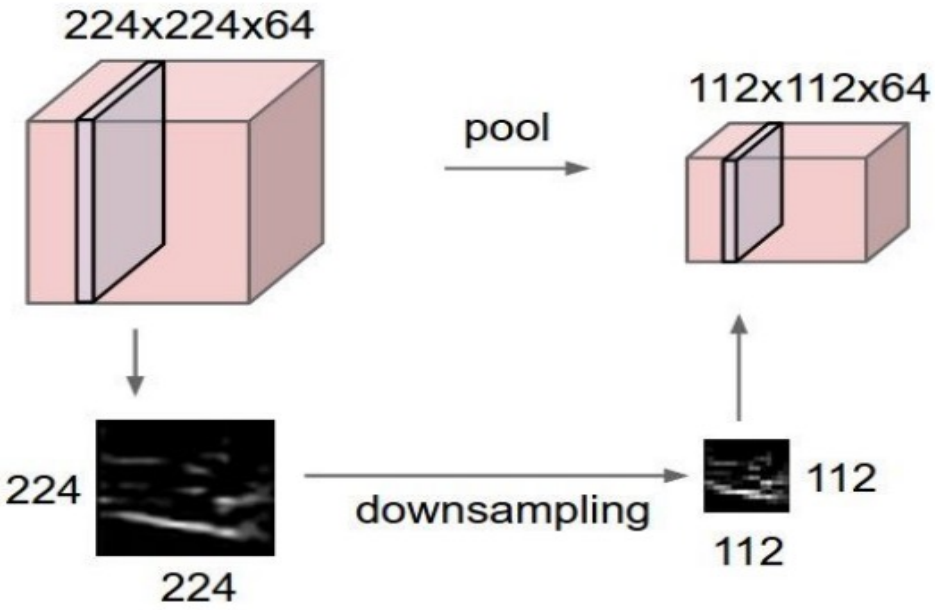
CNNs

- Mạng nơ-ron tích chập là một dãy các lớp tích chập nối liên tiếp nhau xen kẽ bởi các hàm kích hoạt (ví dụ ReLU)

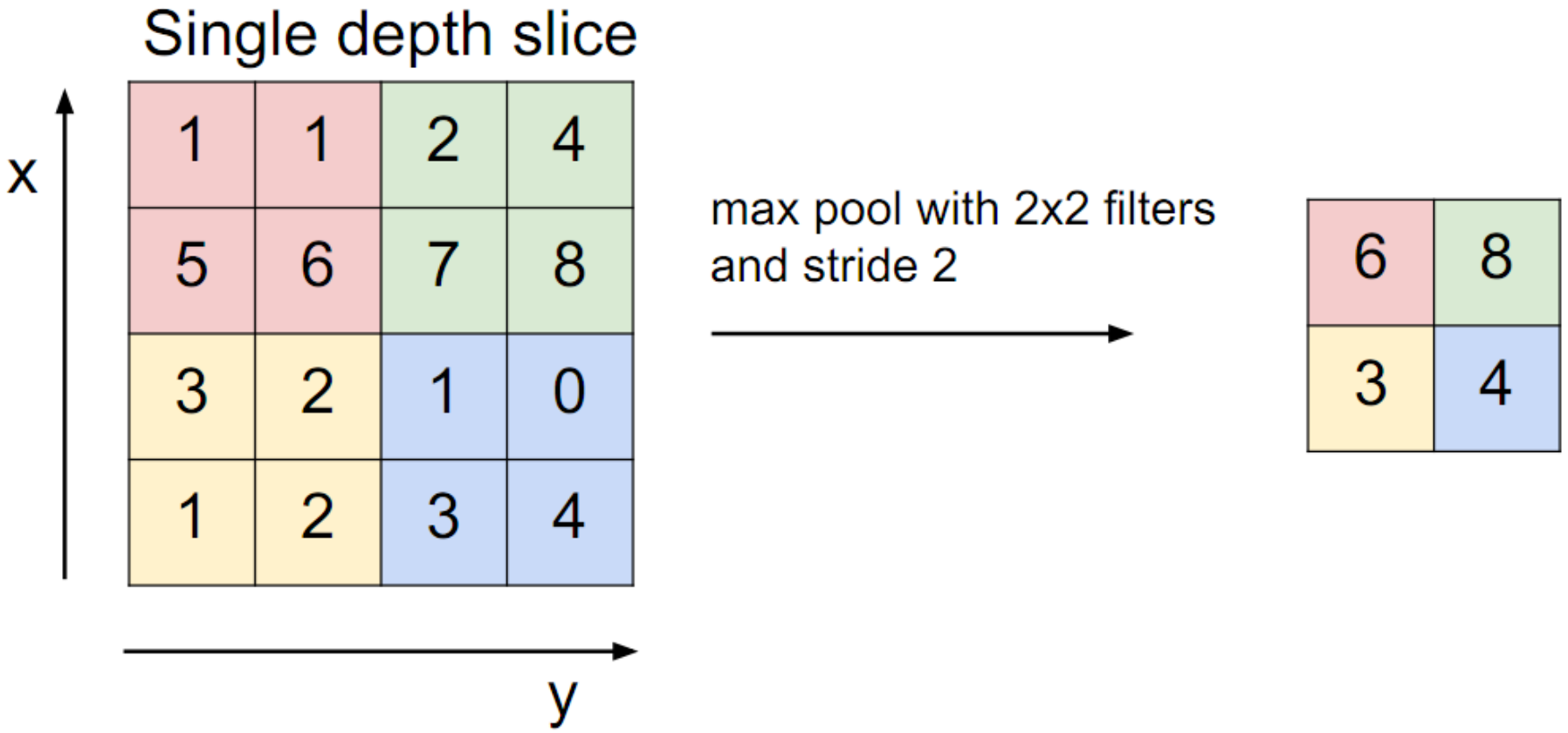


Lớp gộp (pooling layer)

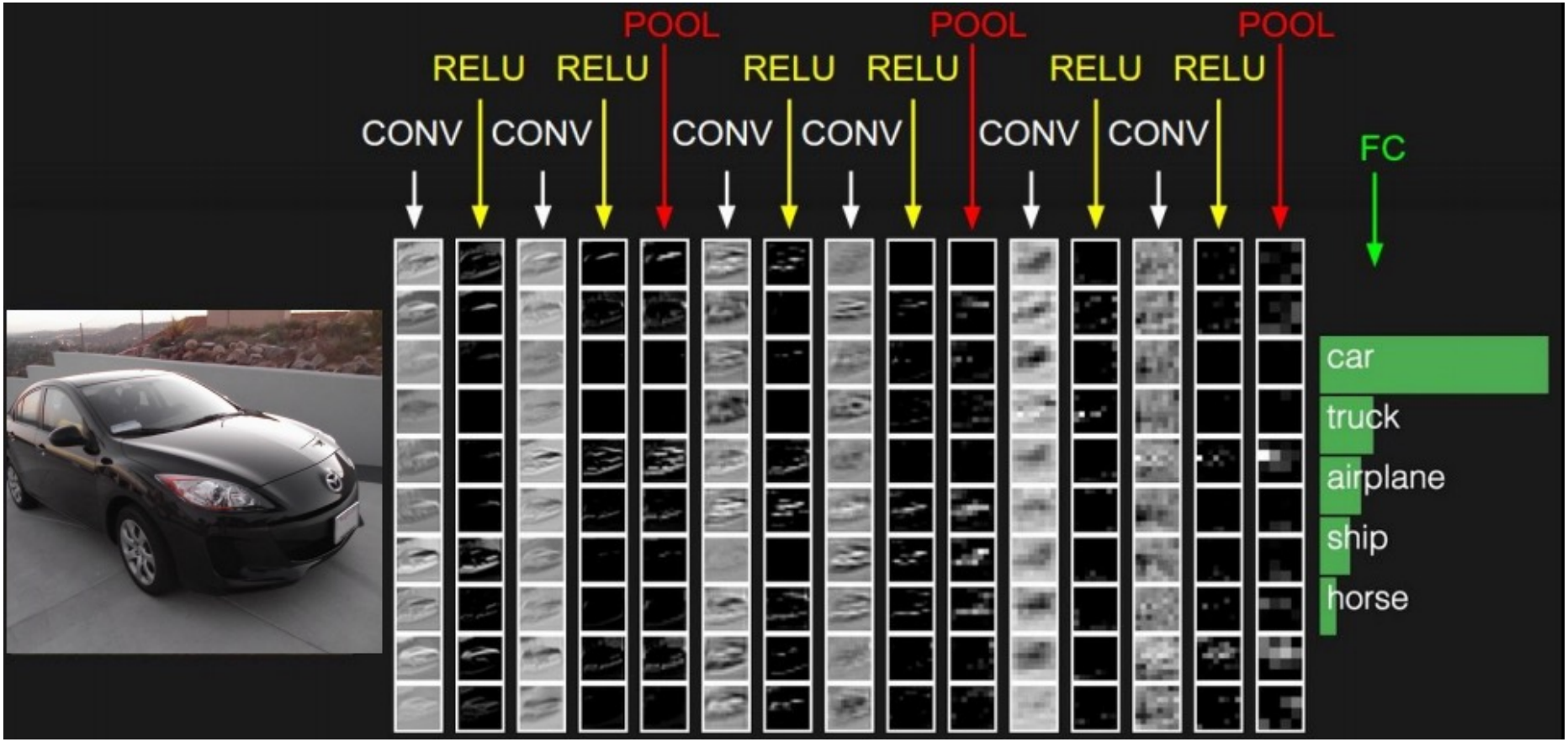
- Giúp giảm độ phân giải của khối dữ liệu để giảm bộ nhớ và khối lượng tính toán
- Hoạt động độc lập trên từng bản đồ kích hoạt
- Lớp gộp max pooling giúp mạng biểu diễn bất biến đối với các thay đổi tịnh tiến (translation invariance) hoặc biến dạng (deformation invariance) của dữ liệu đầu vào



Lớp gộp max pooling



CNNs

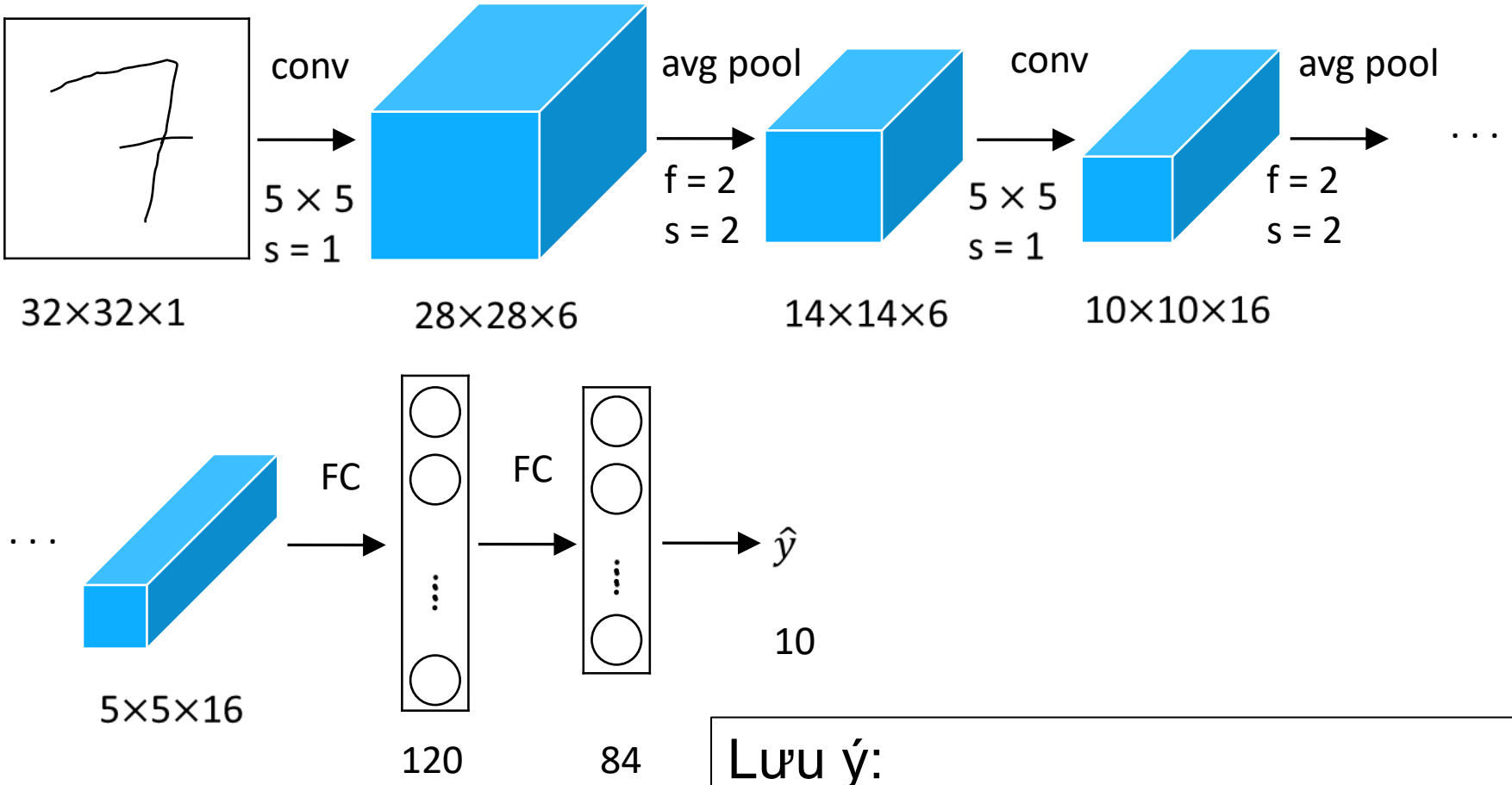

POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Một vài mạng CNN cơ bản

Một số mạng CNNs cơ bản

- LeNet-5
- AlexNet
- VGG
- GoogleNet
- ResNet

LeNet-5



Lưu ý:
Output size = (N+2P-F)/stride + 1

This slide is taken from Andrew Ng

[LeCun et al., 1998]

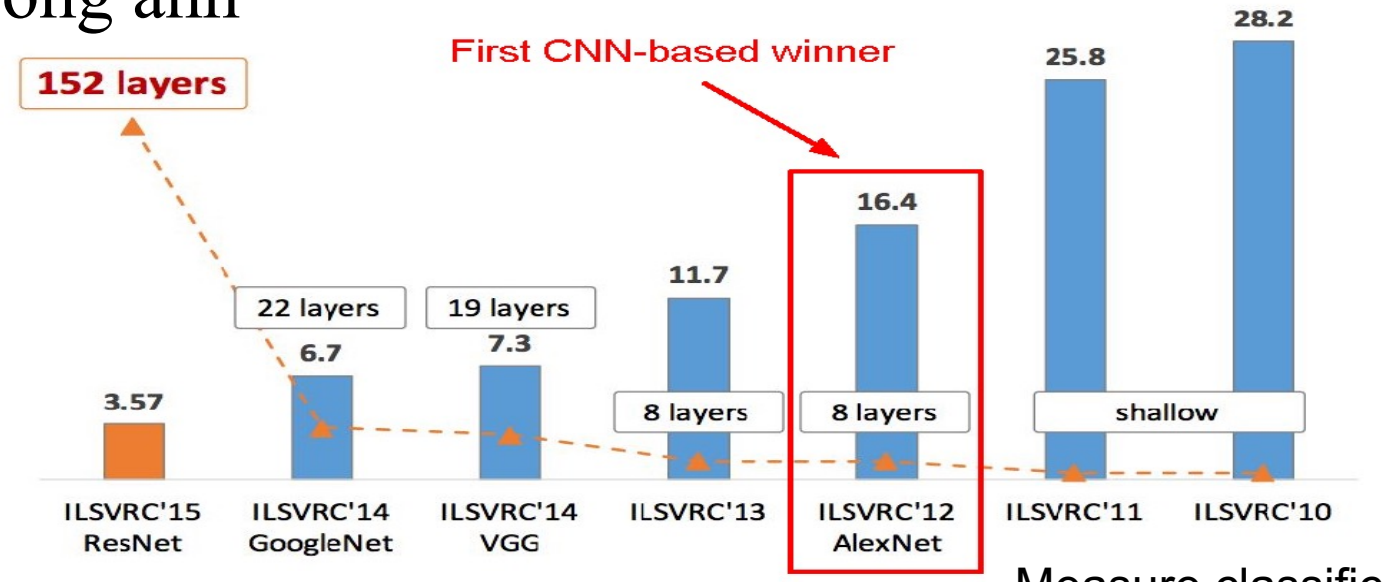
AlexNet

- ImageNet Classification with Deep Convolutional Neural Networks - Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton; 2012
- Một trong những mạng CNNs lớn nhất tại thời điểm đó
- Có 60M tham số số so với 60k tham số LeNet-5

[Krizhevsky et al., 2012]

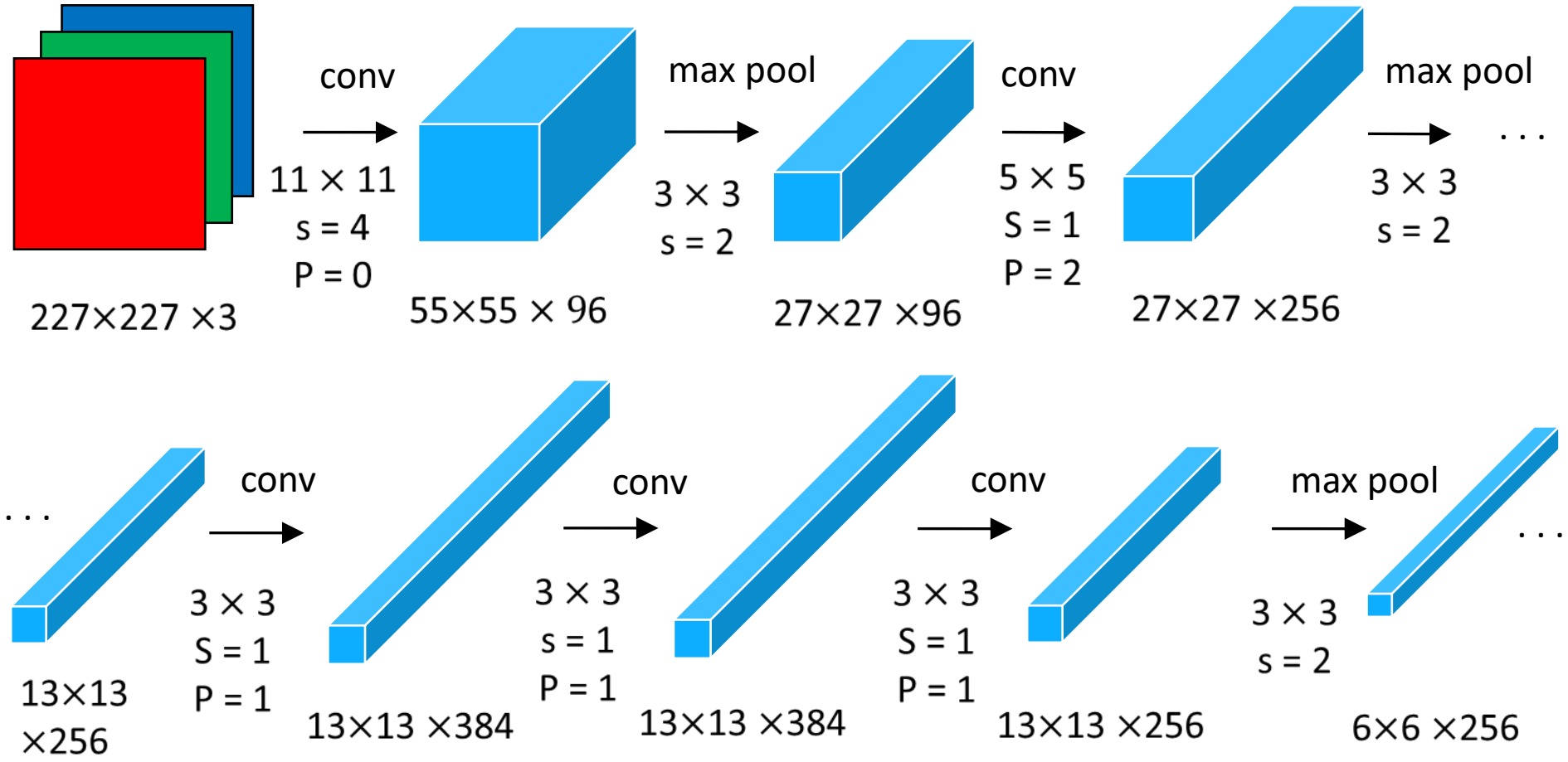
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

- "Olympics" thường niên về lĩnh vực thị giác máy tính.
- Các teams khắp thế giới thi đấu với nhau để xem ai là người có mô hình CV tốt nhất cho các bài toán như phân loại ảnh, định vị và phát hiện đối tượng trong ảnh



Measure classification error

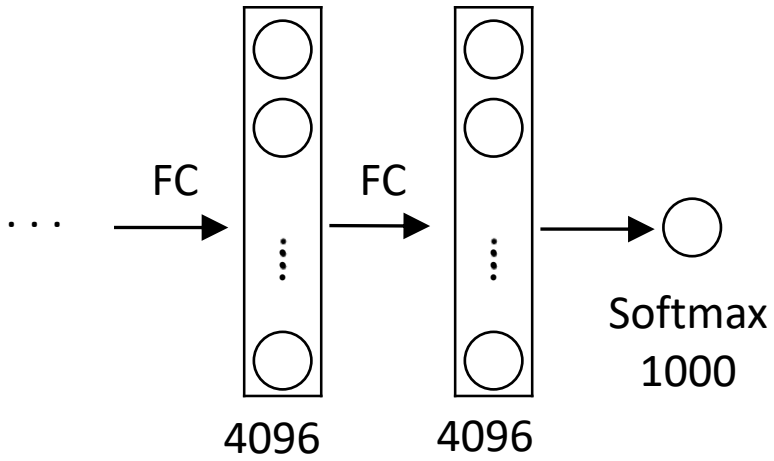
AlexNet



This slide is taken from Andrew Ng

[Krizhevsky et al., 2012]

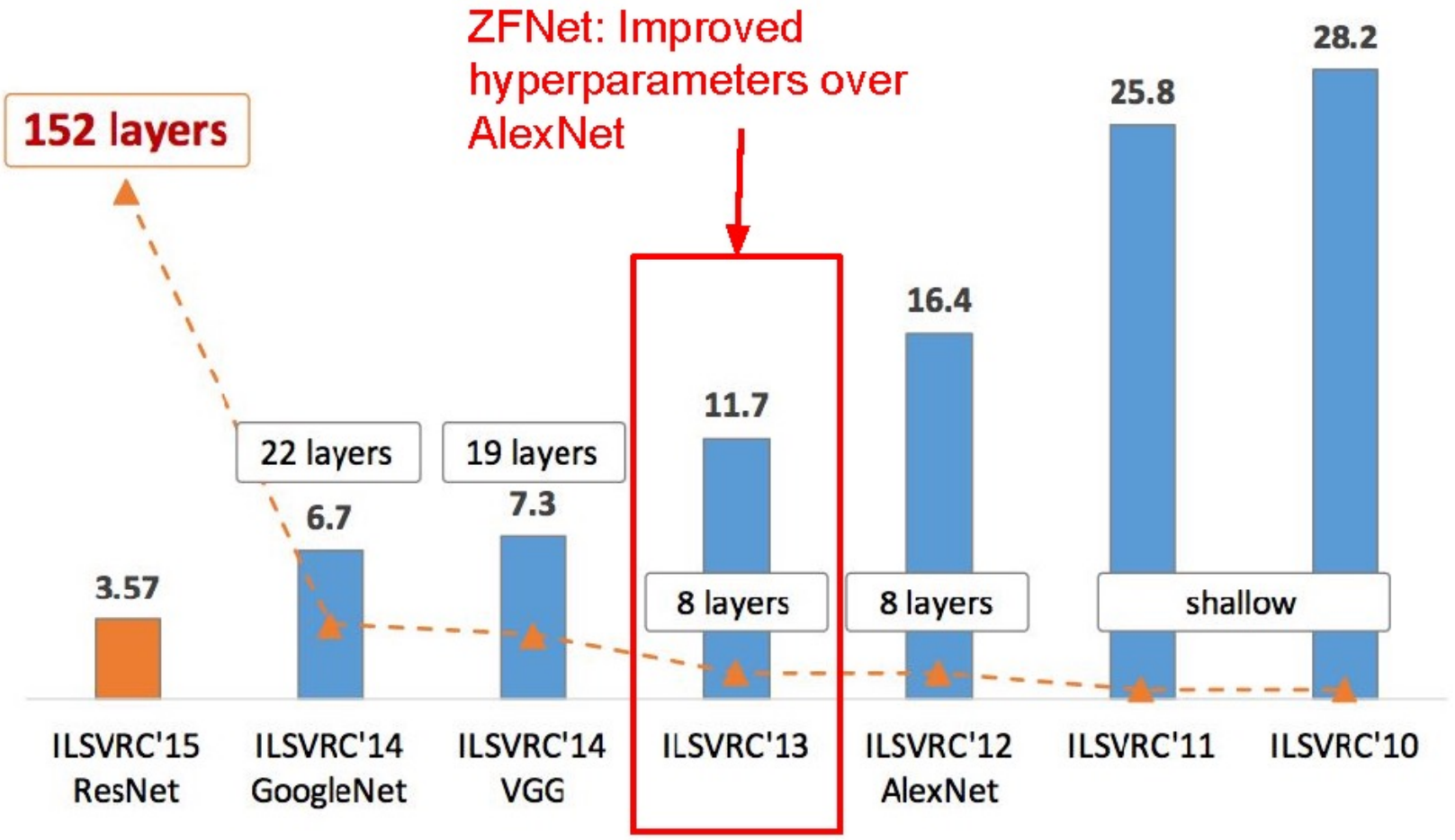
AlexNet



This slide is taken from Andrew Ng

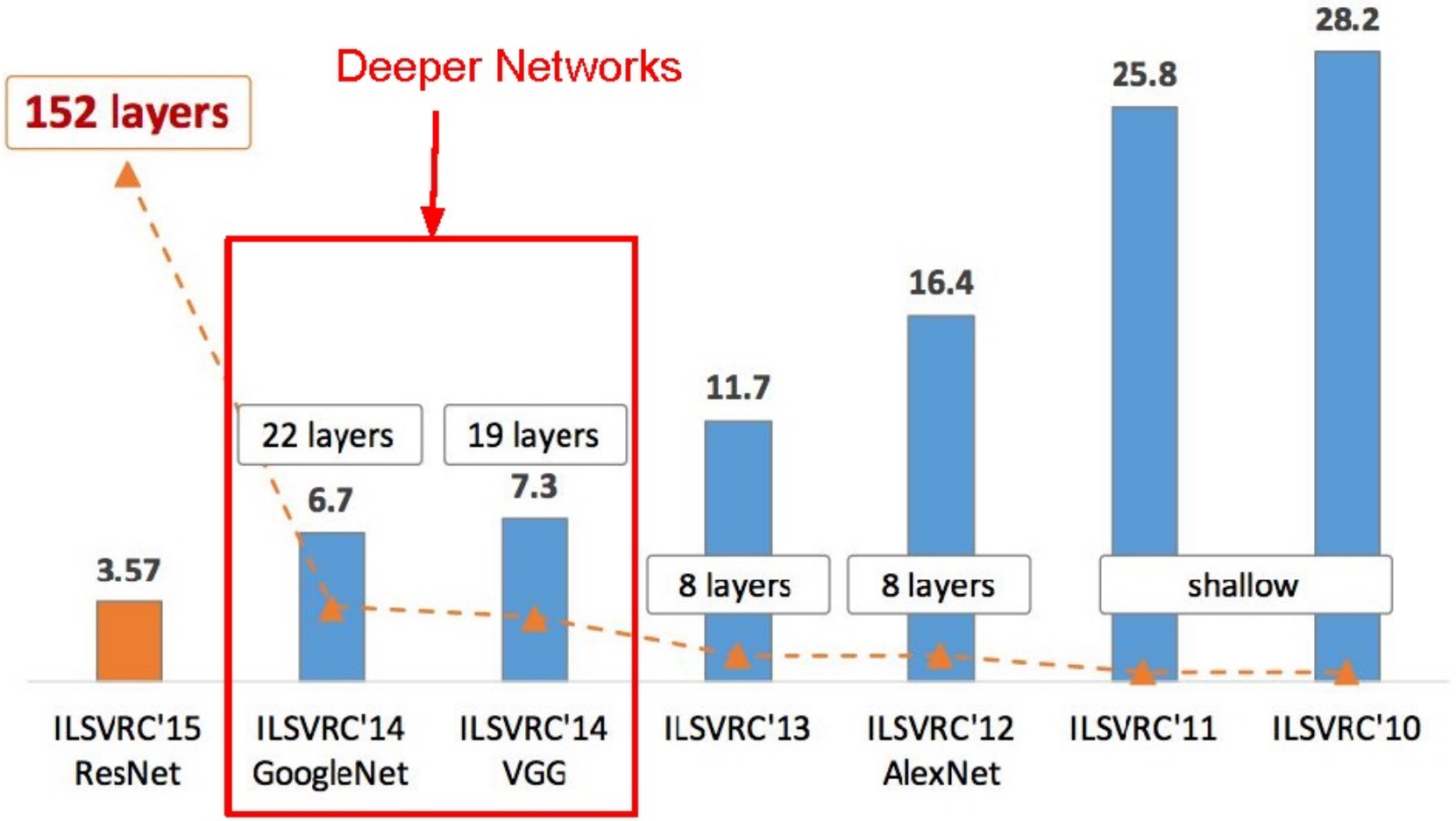
[**Krizhevsky et al., 2012**]

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Slide taken from Fei-Fei & Justin Johnson & Serena Yeung. Lecture 9.

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Slide taken from Fei-Fei & Justin Johnson & Serena Yeung. Lecture 9.

VGGNet

- Very Deep Convolutional Networks For Large Scale Image Recognition - Karen Simonyan and Andrew Zisserman; 2015
- Á quân tại cuộc thi ILSVRC 2014
- Sâu hơn rất nhiều so với AlexNet
- 140 triệu tham số

[Simonyan and Zisserman, 2014]

Input
3x3 conv, 64
3x3 conv, 64
Pool 1/2
3x3 conv, 128
3x3 conv, 128
Pool 1/2
3x3 conv, 256
3x3 conv, 256
Pool 1/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
Pool 1/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
Pool 1/2
FC 4096
FC 4096
FC 1000
Softmax

VGGNet

- Nơ-ron kích thước bé
  Chỉ dùng conv 3x3, stride 1, pad 1
  và 2x2 MAX POOL , stride 2

- Mạng sâu hơn
  AlexNet: 8 lớp
  VGGNet: 16 - 19 lớp

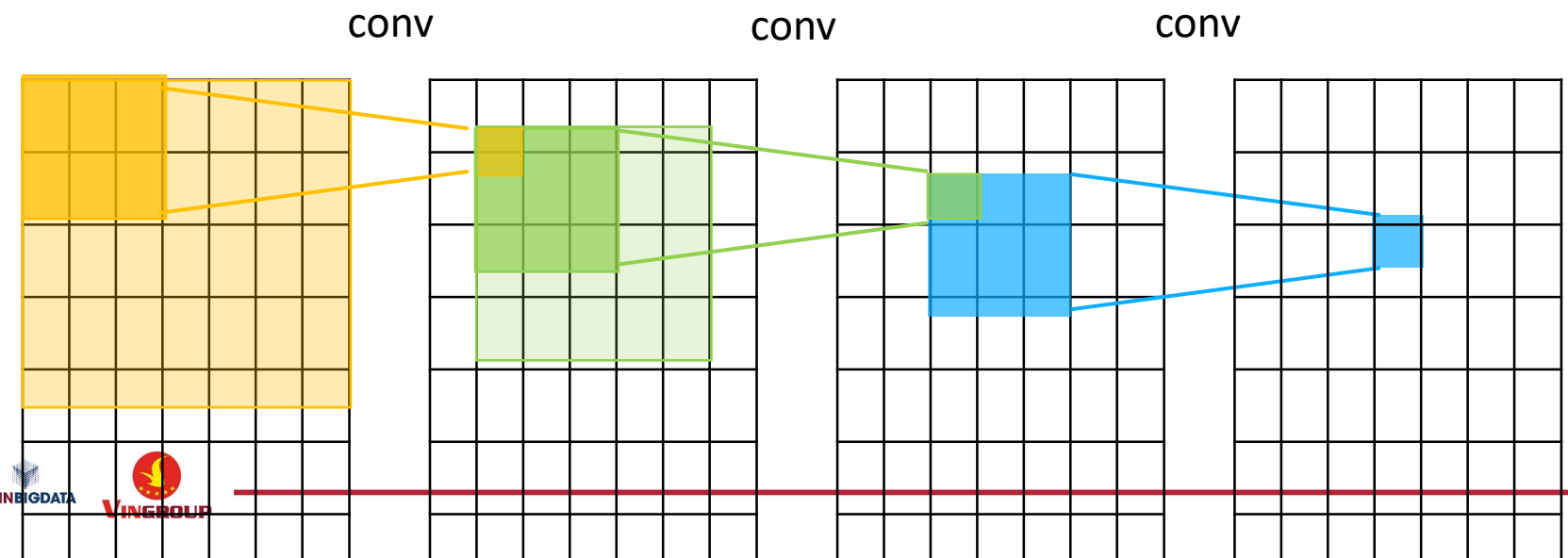
- ZFNet: 11.7% top 5 error in ILSVRC'13
- VGGNet: 7.3% top 5 error in ILSVRC'14

[Simonyan and Zisserman, 2014]

VGGNet

- Tại sao dùng filter bé? (3x3 conv)
- Chồng 3 lớp 3x3 conv (stride 1) có cùng hiệu quả thu nhận thông tin như một lớp 7x7 conv.
- Nhưng sâu hơn, nhiều lớp phi tuyến hơn
- Và ít tham số hơn: 3 * (3^2C^2) vs. 7^2C^2 với C là số kênh của mỗi lớp

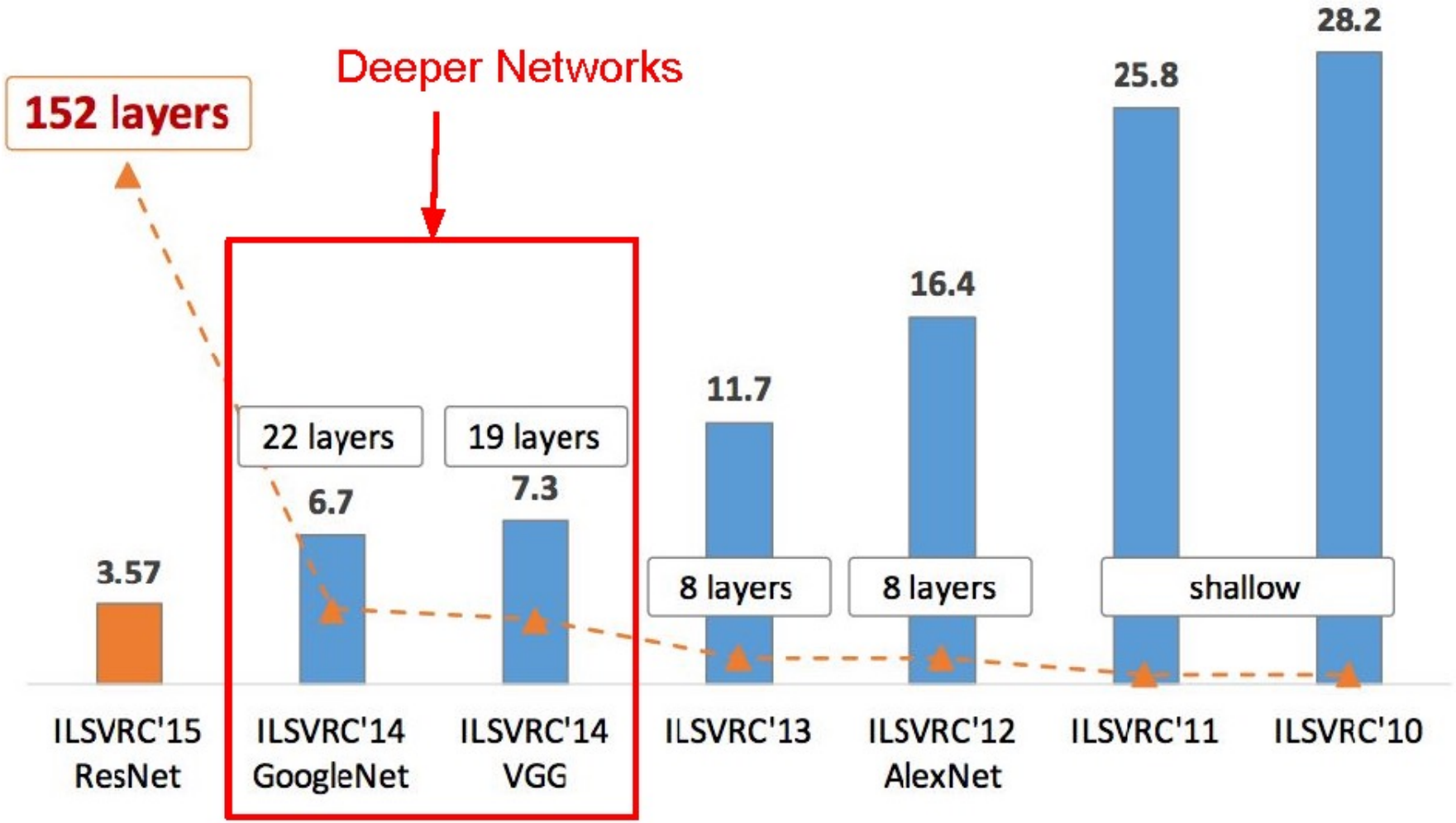
[Simonyan and Zisserman, 2014]



Input	memory: 224*224*3=150K	params: 0
3x3 conv, 64	memory: 224*224*64=3.2M	params: (3*3*3)*64 = 1,728
3x3 conv, 64	memory: 224*224*64=3.2M	params: (3*3*64)*64 = 36,864
Pool	memory: 112*112*64=800K	params: 0
3x3 conv, 128	memory: 112*112*128=1.6M	params: (3*3*64)*128 = 73,728
3x3 conv, 128	memory: 112*112*128=1.6M	params: (3*3*128)*128 = 147,456
Pool	memory: 56*56*128=400K	params: 0
3x3 conv, 256	memory: 56*56*256=800K	params: (3*3*128)*256 = 294,912
3x3 conv, 256	memory: 56*56*256=800K	params: (3*3*256)*256 = 589,824
3x3 conv, 256	memory: 56*56*256=800K	params: (3*3*256)*256 = 589,824
Pool	memory: 28*28*256=200K	params: 0
3x3 conv, 512	memory: 28*28*512=400K	params: (3*3*256)*512 = 1,179,648
3x3 conv, 512	memory: 28*28*512=400K	params: (3*3*512)*512 = 2,359,296
3x3 conv, 512	memory: 28*28*512=400K	params: (3*3*512)*512 = 2,359,296
Pool	memory: 14*14*512=100K	params: 0
3x3 conv, 512	memory: 14*14*512=100K	params: (3*3*512)*512 = 2,359,296
3x3 conv, 512	memory: 14*14*512=100K	params: (3*3*512)*512 = 2,359,296
3x3 conv, 512	memory: 14*14*512=100K	params: (3*3*512)*512 = 2,359,296
Pool	memory: 7*7*512=25K	params: 0
FC 4096	memory: 4096	params: 7*7*512*4096 = 102,760,448
FC 4096	memory: 4096	params: 4096*4096 = 16,777,216
FC 1000	memory: 1000	params: 4096*1000 = 4,096,000

[Simonyan and Zisserman, 2014]

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

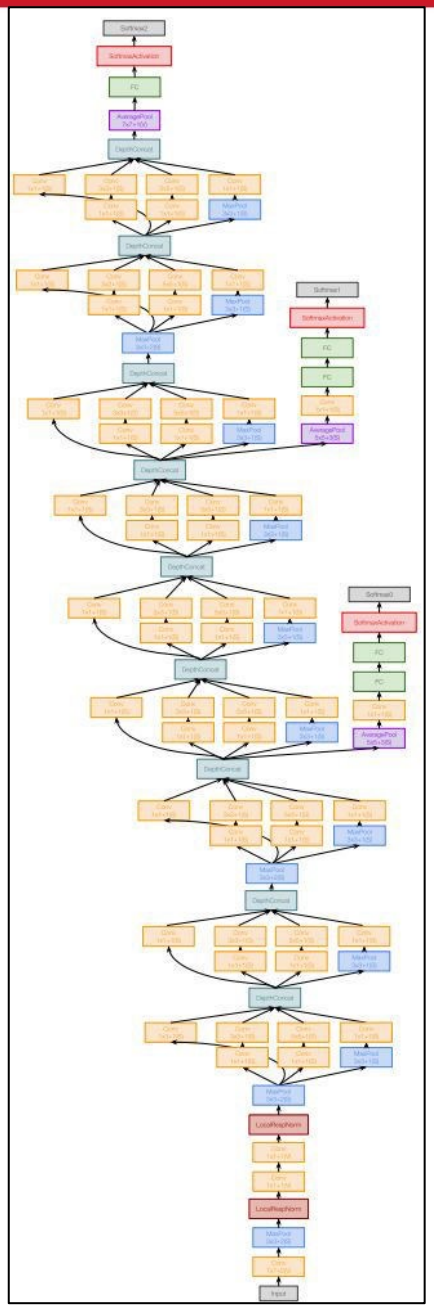


Slide taken from Fei-Fei & Justin Johnson & Serena Yeung. Lecture 9.

GoogleNet

- Going Deeper with Convolutions - Christian Szegedy et al.; 2015
- Vô địch ILSVRC 2014
- Sâu hơn nhiều so với AlexNet
- Số tham số ít hơn 12 lần so với AlexNet
- Tập trung vào giảm độ phức tạp tính toán

[Szegedy et al., 2014]



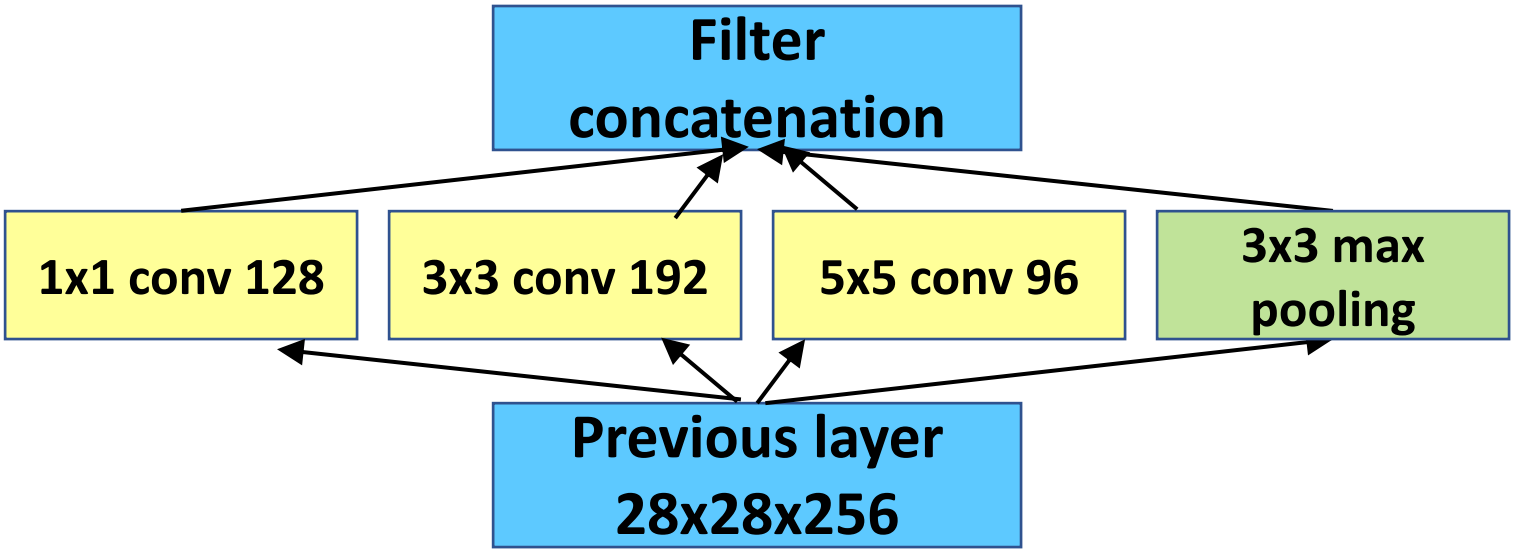
GoogleNet

- 22 lớp
- Khối "Inception"
- Không có lớp kết nối đầy đủ (FC layers)
- Chỉ 5 triệu tham số!
- Vô địch tác vụ phân loại ảnh ILSVRC'14 (6.7% top 5 error)

[Szegedy et al., 2014]

GoogleNet - Naïve Inception Model

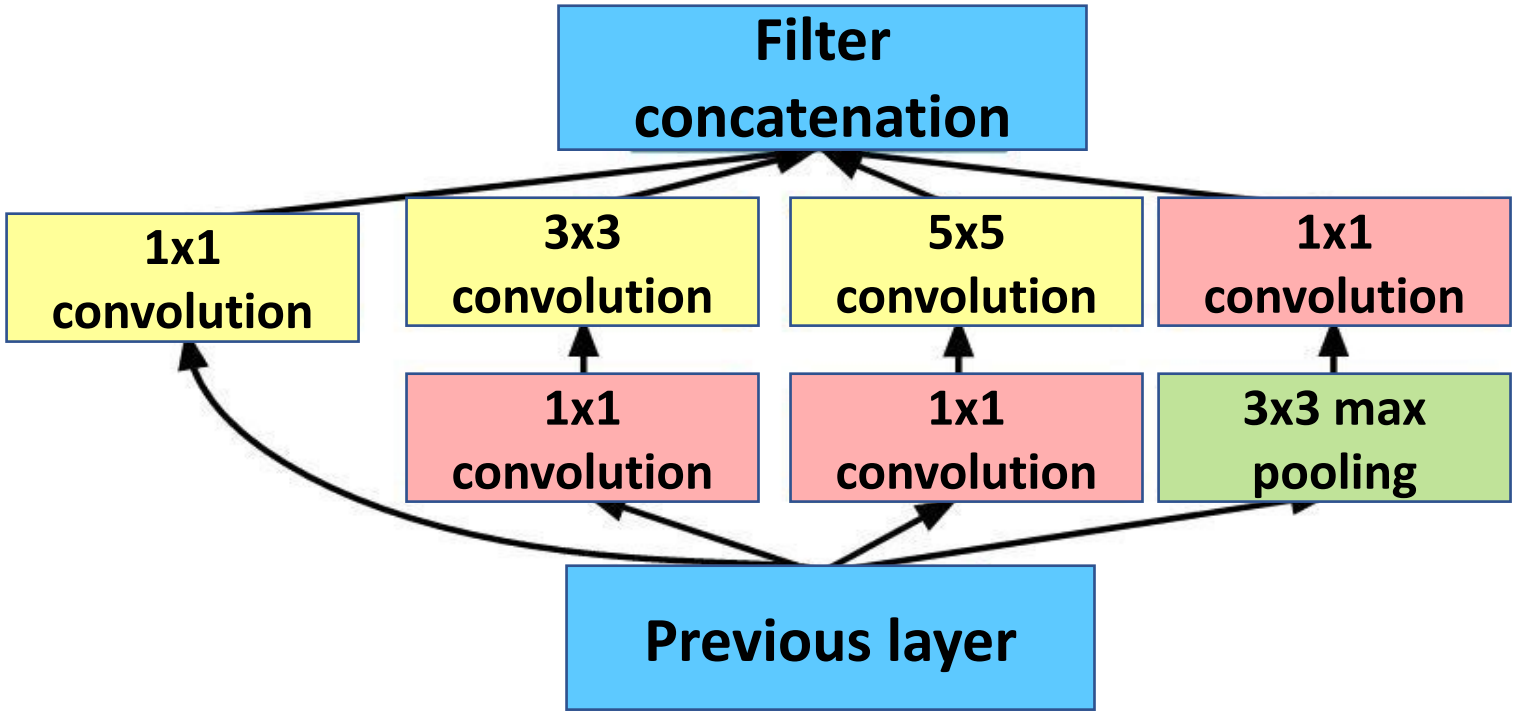
- Số lượng phép tích chập:
- 1x1 conv, 128: 28x28x128x1x1x256
- 3x3 conv, 192: 28x28x192x3x3x256
- 5x5 conv, 96: 28x28x96x5x5x256
- Tổng cộng: 854M ops ==> **Tính toán rất nặng!**



[Szegedy et al., 2014]

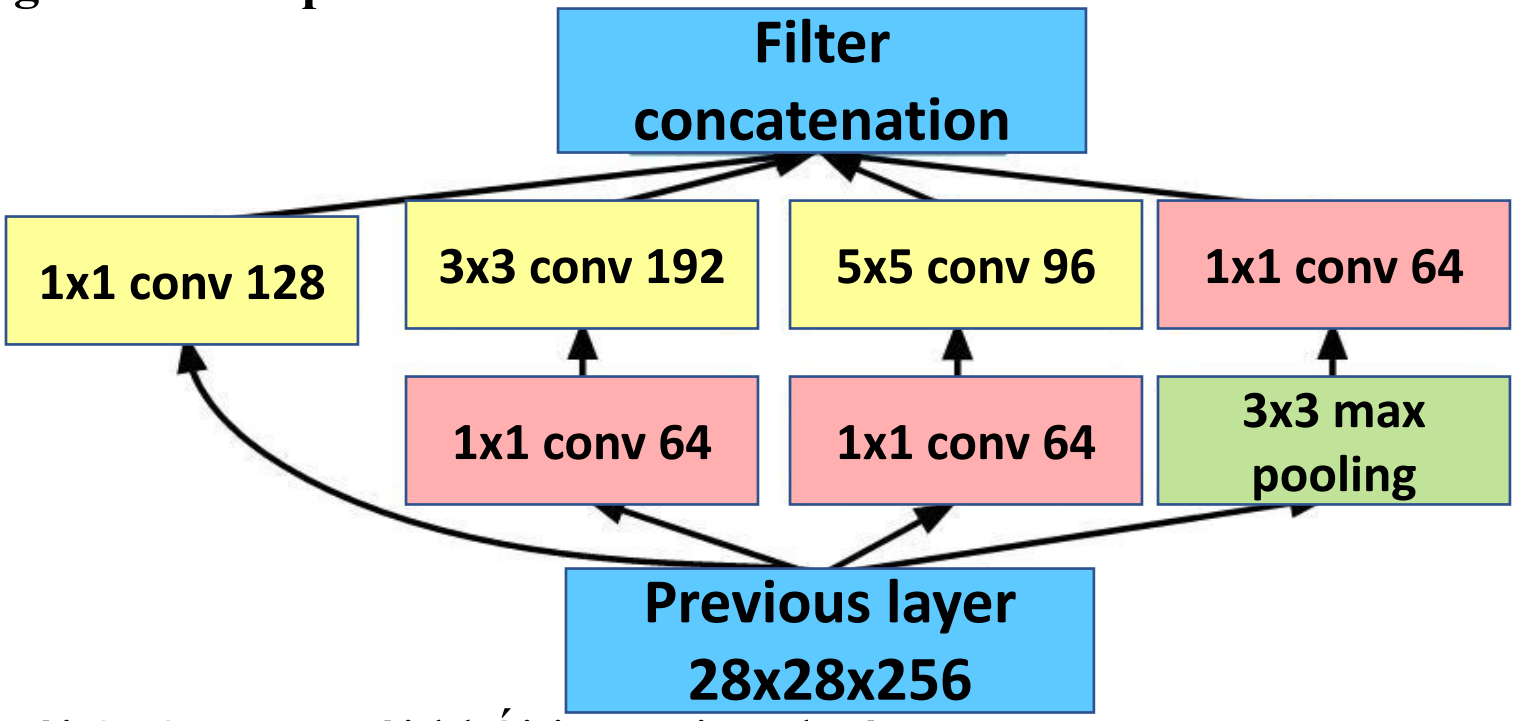
GoogleNet

- Giải pháp: lớp nút cổ chai "bottleneck" sử dụng conv 1x1 để giảm chiều sâu khối dữ liệu.



[Szegedy et al., 2014]

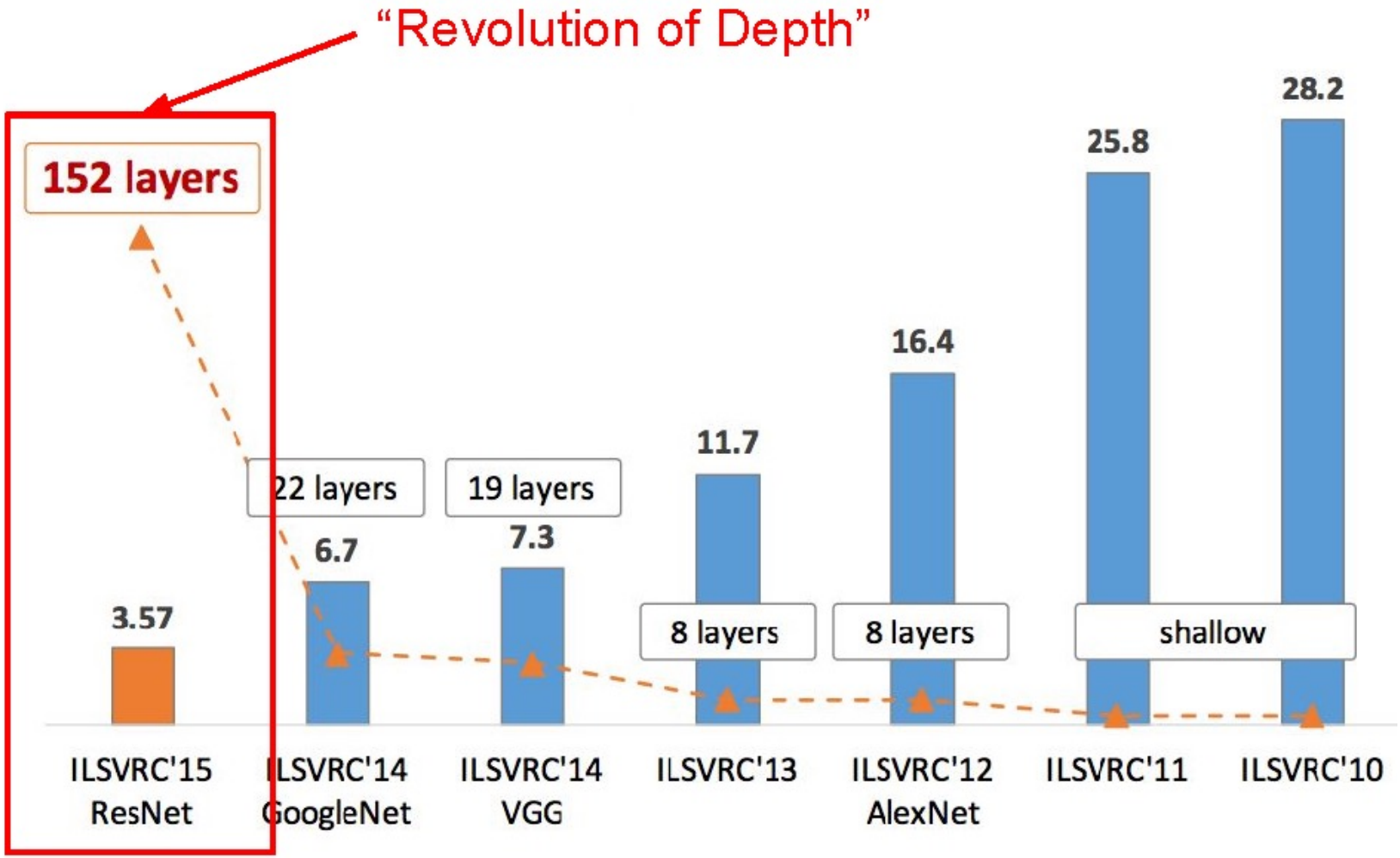
- **Số lượng phép toán tích chập:**
  1x1 conv, 64: 28x28x64x1x1x256
  1x1 conv, 64: 28x28x64x1x1x256
  1x1 conv, 128: 28x28x128x1x1x256
  3x3 conv, 192: 28x28x192x3x3x64
  5x5 conv, 96: 28x28x96x5x5x64
  1x1 conv, 64: 28x28x64x1x1x256
  Tổng số: 353M ops



- So với 854M ops với khối inception thường

[Szegedy et al., 2014]

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

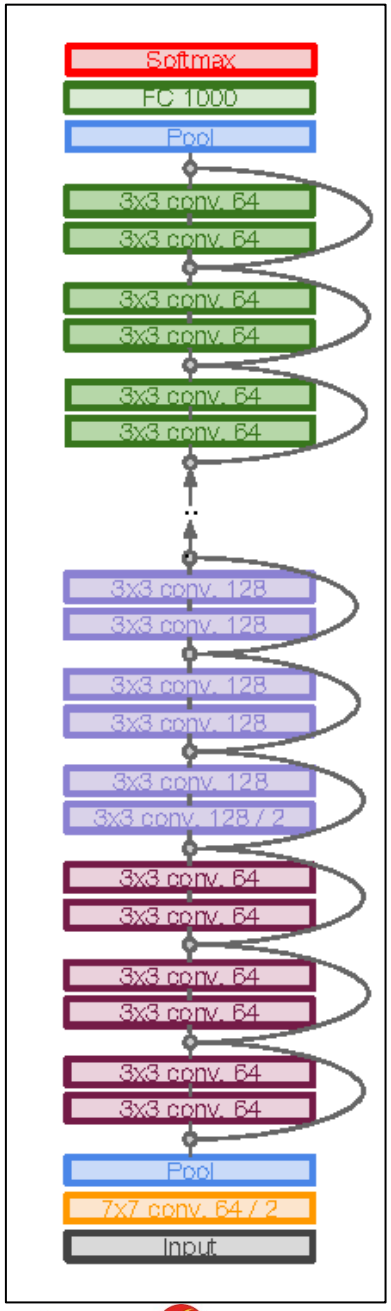


Slide taken from Fei-Fei & Justin Johnson & Serena Yeung. Lecture 9.

ResNet

- Deep Residual Learning for Image Recognition - Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; 2015
- Mạng rất sâu, tới 152 lớp
- Mạng càng sâu càng khó huấn luyện.
- Mạng càng sâu càng chịu nhiều ảnh hưởng của vấn đề triệt tiêu và bùng nổ gradient.
- ResNet đề xuất phương pháp học phần dư (residual learning) cho phép huấn luyện hiệu quả các mạng sâu hơn rất nhiều so với các mạng xuất hiện trước đó.

[He et al., 2015]



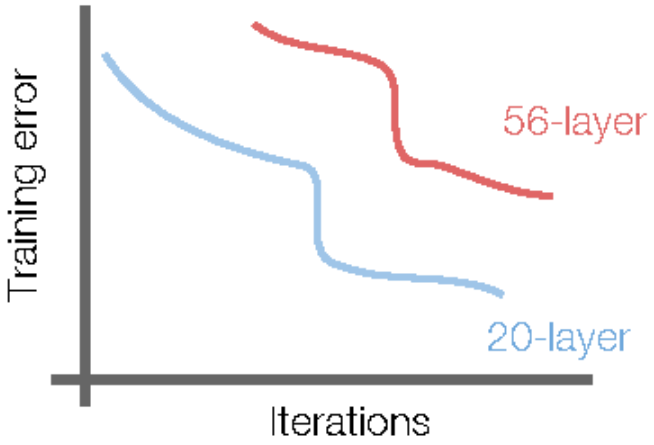
ResNet

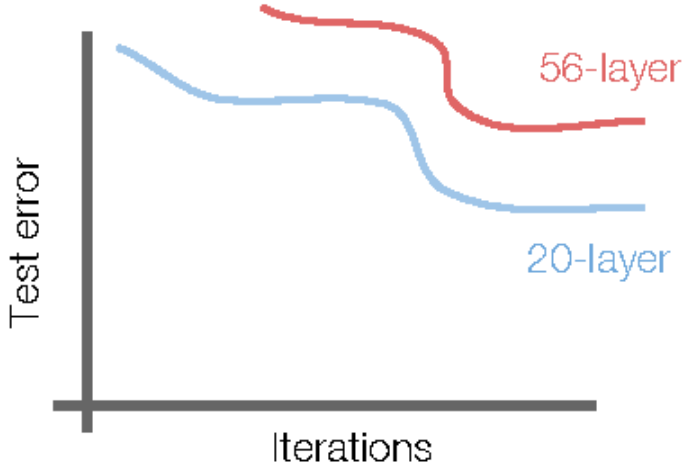
- Vô địch tác vụ phân loại ILSVRC'15 (3.57% top 5 error, trong khi sai số của con người khoảng 5.1%)
- Càn quét tất cả các cuộc thi về phân loại ảnh tại ILSVRC'15 và COCO'15!

[He et al., 2015]

ResNet

- Điều gì xảy ra khi chúng ta tăng độ sâu mạng nơ-ron?
- Mạng 56 lớp làm việc kém hơn cả trên tập huấn luyện lẫn tập test (không phải do overfitting gây ra)

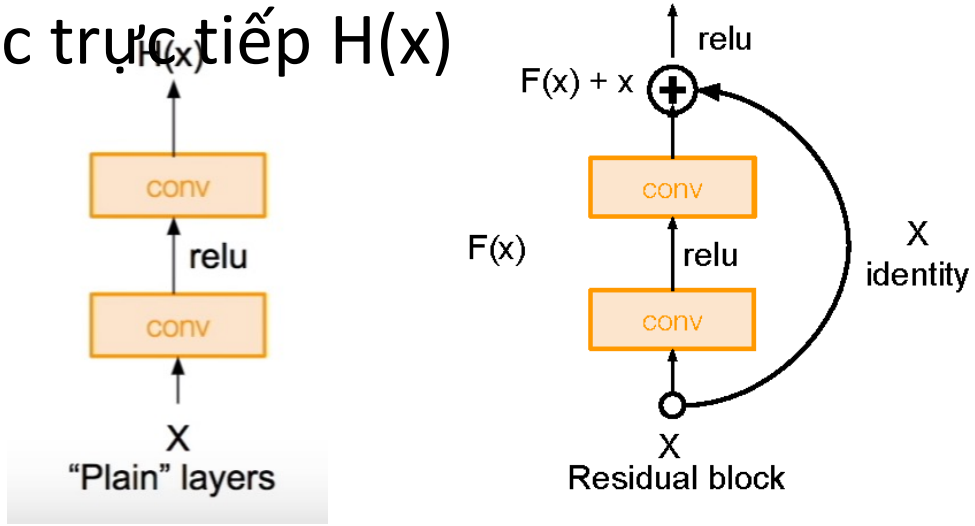




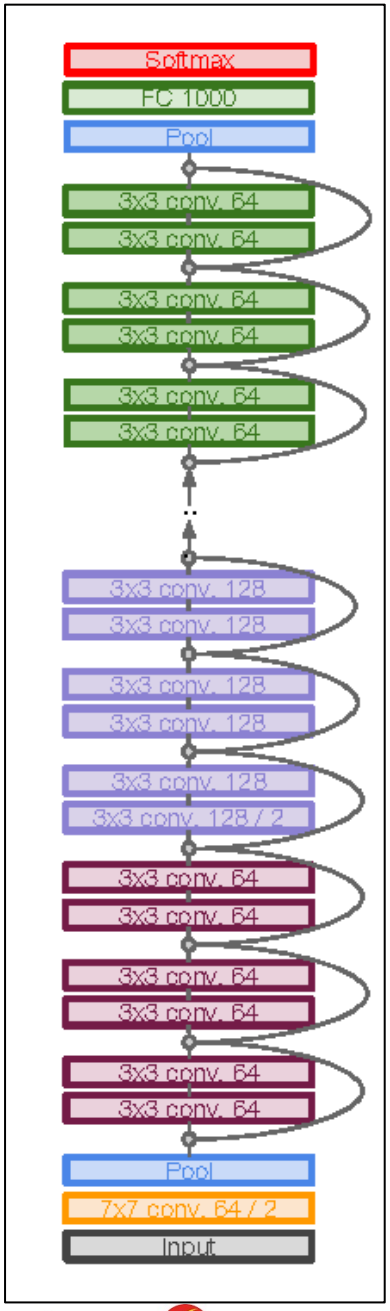
[He et al., 2015]

ResNet

- Giả thiết: Vấn đề ở chỗ bài toán tối ưu. Mạng rất sâu sẽ khó hơn để tối ưu.
- Giải pháp: Dùng các lớp mạng để học biểu diễn phần dư (sự sai khác giữa đầu ra và đầu vào) thay vì học trực tiếp đầu ra như trước.
- Học biểu diễn phần dư F(x) = H(x) – x thay vì học trực tiếp H(x)



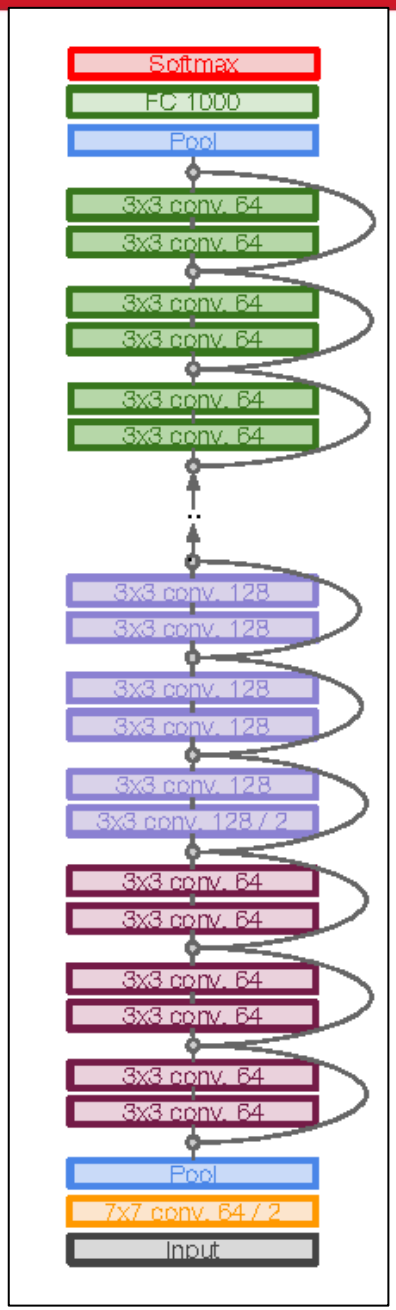
[He et al., 2015]



ResNet

- Kiến trúc ResNet đầy đủ:
- Chồng các khối phần dư residual blocks
- Mỗi khối có hai lớp 3x3 conv
- Định kỳ tăng gấp đôi số lượng filter và giảm độ phân giải bằng conv bước nhảy stride 2
- Lớp conv phụ ở đầu mạng
- Không có lớp FC ở cuối (chỉ có lớp FC 1000 để xuất ra kết quả phân loại 1000 lớp)

[He et al., 2015]

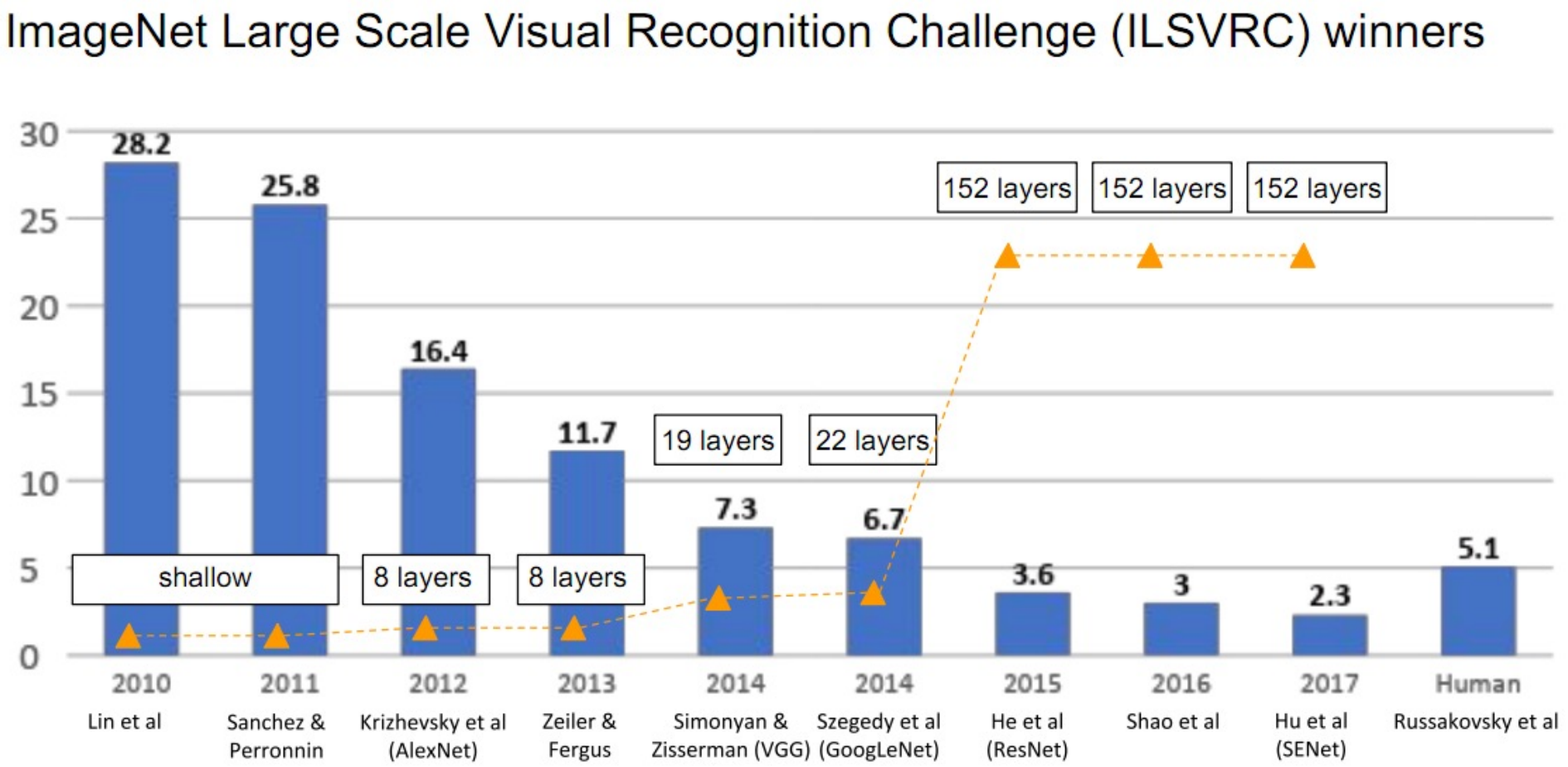


ResNet

- Độ sâu của mạng khi tham gia cuộc thi ImageNet: 34, 50, 101, 152
- Với các mạng sâu (ResNet-50+), tác giả dùng lớp "bottleneck" để tăng hiệu quả (tương tự như GoogLeNet)

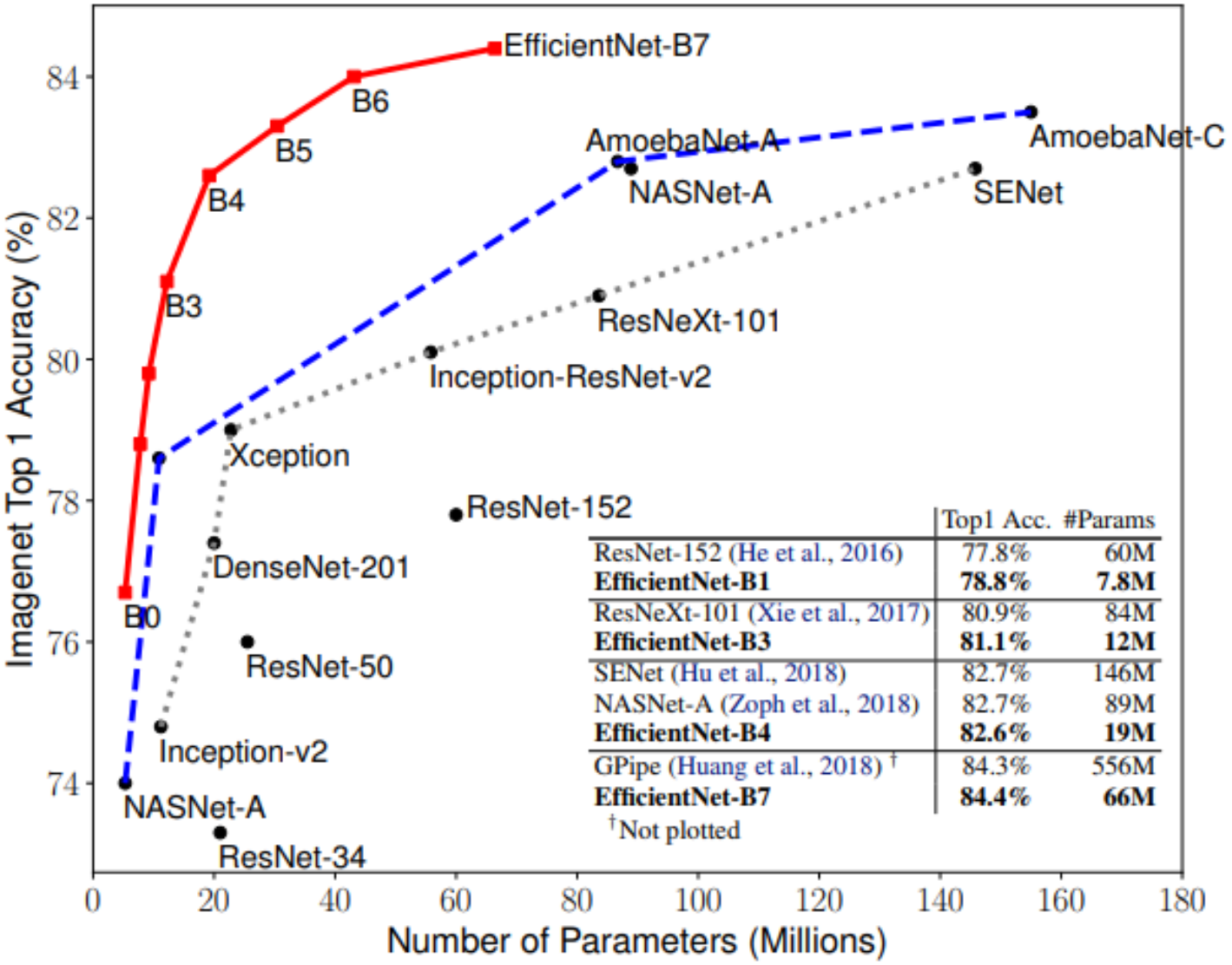
[He et al., 2015]

Recent SOTA



Slide taken from Fei-Fei & Justin Johnson & Serena Yeung. Lecture 9.

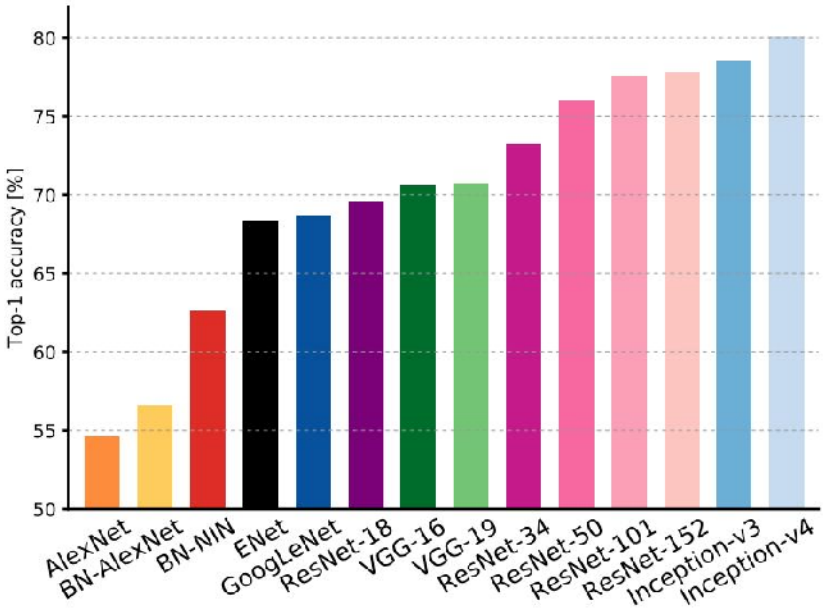
Recent SOTA

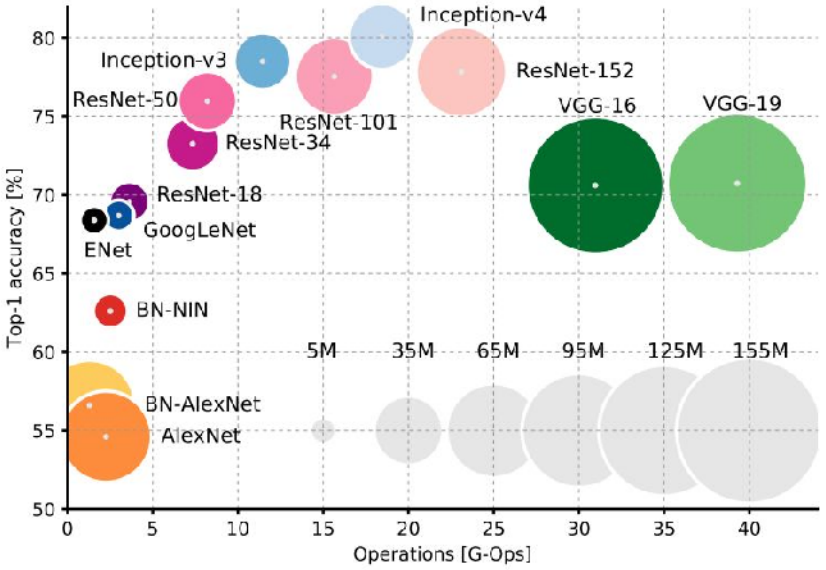


	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Accuracy comparison





Tài liệu tham khảo

1. Khóa học Intro to DL của MIT:

http://introtodeeplearning.com/

2. Khóa học cs231n của Stanford:
http://cs231n.stanford.edu/